



THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Reducing the Risks of Telehealthcare Expansion Through the Automation of Efficiency Evaluation

Cristina Adriana Alexandru



Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2015

Abstract

Several European countries, including the UK, are investing in large-scale telehealthcare pilots, to thoroughly evaluate the benefits of telehealthcare. Due to the high level of risk associated with such projects, it becomes desirable to be able to predict the success of telehealthcare systems in potential deployments, in order to inform investment and help save resources. An important factor for the success of any telehealthcare deployment is usability, as it helps to achieve the benefits of the technology through increased productivity, decreased error rates, and better acceptance. In particular, efficiency, one of the characteristics of usability, should be seen as a central measure for success, as the timely care of a high number of patients is one of the important claims of telehealthcare. Despite the recognized importance of usability, it is seen as secondary in the design of telehealthcare systems. The resulting problems are difficult to predict due to the heterogeneity of deployment contexts.

This thesis proposes the automation of usability evaluation through the use of modelling and simulation techniques. It describes a generic methodology which can guide a modeller in reusing models for predicting characteristics of usability within different deployment sites. It also describes a modelling approach which can be used together with the methodology, to run in parallel a user model, inspired from a cognitive architecture, and a system model, represented as a basic labelled transition system. The approach simulates a user working with a telehealthcare system, and within her environment, to predict the efficiency of the system and work process surrounding it. The modeller can experiment with different inputs to the models in terms of user profile, workload, ways of working, and system design, to model different potential- real or hypothetical- deployments, and obtain efficiency predictions for each. A comparison of the predictions helps analyse the effects on efficiency of changes in deployments.

The work is presented as an experimental investigation, but emphasises the great potential of modelling and simulation for helping to inform investment, help reduce costs, mitigate risks and suggest changes that would be necessary for improving the usability, and therefore success of telehealthcare deployments. My vision is that, if used commercially, the approaches presented in this thesis could help reduce risks for scaling up telehealthcare deployments.

Lay Summary

Telehealthcare has the potential of addressing current and future demographic challenges faced by healthcare systems worldwide by aiding in the provision of better patient care at lower costs. Several European countries, including the UK, are investing in large scale telehealthcare pilots to thoroughly assess its feasibility.

This thesis proposes methods for automating the otherwise costly usability evaluation of telehealthcare systems within different deployment contexts, which could help such pilots assess usability as a factor of success, to inform investment and mitigate risks. It describes a set of steps for building, validating and reusing models to help predict characteristics of usability within different, real or hypothetical, deployments. It also proposes a modelling approach which could be used to predict efficiency as a characteristic of usability. The approach allows the representation (i.e. modelling) of a user of the telehealthcare system and of the system itself, the simulation of the user's work on the system and in her environment, and the computation of total work times as an indication of the efficiency of the system and work process. The models of the user and system can be changed to reflect differences in workload, user profile, ways of working or system design within different deployments, to investigate efficiency there, and how these changes would affect it. Such an investigation could help reduce costs in time and resource allocation in existing deployments, assess whether work with the telehealthcare system would be manageable within proposed new deployments, and inform what would need to be changed if not. It also allows the hypothetical exploration of the effects on efficiency of foreseeable changes to existing deployments, thus helping to identify and mitigate risks.

If used commercially, the approaches presented in this thesis could constitute a real asset for investors, managers and software developers in the implementation and scaling up of telehealthcare systems.

Acknowledgements

First and foremost, I would like to thank my supervisors, Prof. Perdita Stevens, Prof. Stuart Anderson and Prof. Brian McKinstry, for their consistent support. I am very grateful to Prof. Perdita Stevens for her continuous help, reliability and constructive criticism, which have definitely helped me improve, both in my research and as a person. I would like to thank Prof. Stuart Anderson for always encouraging me, and for his valuable practical suggestions. I am very grateful to Prof. Brian McKinstry for his insights from his vast experience in telehealthcare, for involving me in the evaluation of telehealthcare systems and for believing in the usefulness of my work.

This thesis would not have been possible without the valuable contribution of the call operator team from NHS 24, and of the clinicians from The Edinburgh Community Respiratory Team, The East and Midlothian COPD Team, The West Lothian Intensive Case Management Team, the Lothian Unscheduled Care Service team, and the Heart Failure Nurse team. I thank them for their openness and valuable insights regarding their work. A special thanks to Dr. John Steyn, for inviting me to evaluate the telehealthcare system used by the teams, and for appreciating my work.

I would like to thank my friend, Mihaela Dragomir, for taking the time to read my lengthy thesis and send me detailed comments. I would also like to thank Aurora Constantin, for the interesting talks and continuous support.

Finally, I would like to thank my loved ones. I thank my boyfriend, Johnny, for always believing in me and for being there for me during the ups and downs of my experience as a PhD student. I thank my mother for her continuous encouragement and understanding, for sharing from her experience and for her enormous support at a distance. I thank my father for his love and for being proud of any of my accomplishments. I thank my younger brother, Tudor, for his humour and for inspiring me by his passion for his work. I thank my elder brother, Daniel, for seeing me take a PhD even before starting it. This thesis would not have been possible without you all!

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Previous experience described in [25, 27] motivated the choice of subject for this thesis, which is described in Chapters 1 and 2. The data and qualitative observations which informed and constituted inputs for the approach presented in this thesis, as described in Chapters 3 and 8, were collected by me during a usability study of a telehealthcare system performed in Lothian, Scotland, under the supervision of Dr. John Steyn from NHS Lothian. The basic ideas behind the proposed approach were first described in [28], and then summarised, enhanced and showcased with different examples in [29] and [26]. The methodology presented in Chapter 4 is a generalised extension of the one described in these papers. The papers presented the different stages of the development of the modelling approach described in Chapter 6 and Appendix E. The first two of these papers were written jointly with Prof. Perdita Stevens.

(Cristina Adriana Alexandru)

To my parents

Contents

1	Introduction	1
1.1	Current Challenges in Healthcare	1
1.2	A Potential Technological Solution	2
1.3	Terminology Considerations	3
1.4	The Telehealthcare Services Under Consideration in This Thesis . . .	4
1.5	Is Telehealthcare Feasible?	5
1.6	European Projects Scaling up Telehealthcare	7
1.7	Thesis Claim	9
1.8	Aims and Objectives	13
1.9	Summary of the Thesis	14
2	Literature Review	17
2.1	Overview	17
2.2	The Issue of Usability	17
2.2.1	Definitions and Characteristics of Usability	17
2.2.2	Benefits of Usability	21
2.3	Considerations of Efficiency in This Thesis	23
2.4	Introduction to Usability (and Efficiency) Evaluation	24
2.5	Usability (and Efficiency) Evaluation Methods	25
2.5.1	Empirical Methods	25
2.5.1.1	Testing Methods	26
2.5.1.2	Inquiry Methods	28
2.5.1.3	Drawbacks of Involving Users in Usability Evaluation	29
2.5.2	Analytical Methods	30
2.5.2.1	Usability Inspection Methods	30
2.5.2.2	Model-Based Evaluation Methods	33
2.5.2.2.1	GOMS	34

2.5.2.2.2	PUM	36
2.5.2.2.3	Cognitive Architectures	38
2.5.2.2.3.1	Soar	38
2.5.2.2.3.2	ACT-R	41
2.5.2.2.3.3	ICARUS	44
2.5.2.2.4	Choosing a Model-Based Evaluation Method	48
2.6	Practice of Usability Evaluation for Medical Designs	49
2.7	Other Related Work	50
2.7.1	From Systems Engineering- Modelling and Simulation	50
2.7.2	From Scientific Management	52
2.7.3	From Project Management	53
3	Usability Study of a Telehealth System Used in Lothian, Scotland	55
3.1	Overview	55
3.2	The Telehealthcare Service	56
3.3	More Details about the Telehealthcare System	57
3.4	Purposes of the Study	61
3.5	Participant Recruitment	64
3.6	Data Collection Methods	64
3.7	Data Analysis Methods	66
3.8	Results	68
3.8.1	Demographics	68
3.8.2	Support	69
3.8.3	Navigation Problems	70
3.8.3.1	Slow Navigation Within the Website	70
3.8.3.1.1	General Comments	70
3.8.3.1.2	Specific Navigation Problems and User Sug- gestions	71
3.8.3.1.2.1	Incomplete or Inappropriate Sort- ing Options	71
3.8.3.1.2.2	Information Inconsistencies, Com- pounded With Problem 1	72
3.8.3.1.2.3	Needed Functionality Being Diffi- cult to Reach	72

3.8.3.1.2.4	Monitoring Work Information Not Being Available Where Required .	74
3.8.3.1.2.5	Problem 4 Compounded With Problem 1	79
3.8.3.2	Slow Navigation Between System Applications Due to a Lack of Integration	80
3.8.3.2.1	General Comments	80
3.8.3.2.2	Suggestions	81
3.8.3.2.2.1	Website-Spreadsheet Integration for COs	83
3.8.3.2.2.2	Website-Spreadsheet Integration for Care Providers	84
3.8.3.2.2.3	Website-Email Facility Integration for COs	85
3.8.3.2.2.4	Integration Between the Email Facility and the Telemonitoring Website for Care Providers	85
3.8.3.2.2.5	Website-Statistics Spreadsheet Integration for COs	87
3.8.4	Presentation Problems	87
3.8.4.1	Problems With the Layout of the Website Pages . .	87
3.8.4.1.1	Needed Information Only Being Reachable by Using Scroll Bars	88
3.8.4.1.2	Use of a Fixed Width Layout	89
3.8.4.1.3	Buttons Only Being Accessible by Using Scroll Bars/ The Website Requiring a Higher Resolution than the Users Have Available on their Screens	90
3.8.4.1.4	Counter-intuitive Button Placement	91
3.8.4.2	Data Representation Problems	92
3.8.4.2.1	Inappropriate Ordering or Structuring of Information in Tables	92
3.8.4.2.2	Inappropriate Highlighting of Information	94
3.8.4.2.3	Inappropriate Colouring for Distinguishing Different Information	98

3.8.4.3	Problems with Button Names	98
3.8.5	Content Problems	101
4	The Methodology	107
4.1	Overview	107
4.2	Basic Description of the Methodology	107
4.3	The Reference Deployment Site	108
4.4	Real Potential Deployment Sites	109
4.5	Hypothetical Potential Deployment Sites	110
4.6	Advantages of the Methodology	111
4.7	Preconditions for the Optimal Use of the Methodology	112
5	An Initial Modelling Approach	115
5.1	Overview	115
5.2	Initial Choice of a Modelling Approach	116
5.3	Description of the Initial Modelling Approach	117
5.3.1	Overview	117
5.3.2	Notions	117
5.3.3	Behaviour	121
5.3.4	Types of Problems Detected by the Framework	125
5.4	Disadvantages of the Initial Modelling Approach	127
6	An Improved Modelling Approach	129
6.1	Overview	129
6.2	Inspiration from a Cognitive Architecture	129
6.3	Advantages of ICARUS for the Purposes of this Thesis	130
6.4	Description of the New Modelling Approach	131
6.4.1	The Structure of the Models	131
6.4.1.1	Overview	131
6.4.1.2	Notions for the User Model	131
6.4.1.3	Notions for the Device Model	136
6.4.2	The Behaviour of the Models	137
6.4.2.1	Overview	137
6.4.2.2	Perception	138
6.4.2.3	Conceptual Inference	138
6.4.2.4	Skill execution	139

6.4.2.5	Mechanisms for Remembrance/Forgetfulness . . .	141
6.4.2.6	Behaviour of the Device Model	144
6.4.2.7	Behaviour of the Controller	145
6.4.3	Inputs	145
7	Engineering an Application for Running Simulations	147
7.1	Overview	147
7.2	Design of the Java Application Implementing the Modelling Approach	147
7.3	System Requirements for the Java Application	148
7.4	Modeller Inputs Required As Prerequisites for Running Simulations .	148
7.5	How it All Works	151
7.5.1	Generating the Inputs Required for the Simulation Run	152
7.5.2	Running the Java Simulation	153
7.5.3	Generating Time Distribution Charts	154
7.5.4	Calculating Time Distributions	154
7.5.5	Putting It All Together	154
8	Data Collection from a Case Study of a Telehealthcare System in Use	157
8.1	Overview	157
8.2	Purposes of the Study for My Modelling Approach	157
8.3	Data Collection Methods	159
8.4	Data Analysis Methods	159
8.5	Data Collection Limitations	161
8.6	Choice of Users to Inspire User Models	162
8.7	Building Inputs for the User and Device Models	163
9	Evaluation of the Methodology and Modelling Approach	169
9.1	Overview	169
9.2	Basic Case Study	170
9.3	First Scenario- Effects of User Experience and Ways of Working . . .	172
9.3.1	Addressing Question 1: Analysis of the Current Context . . .	172
9.3.2	Addressing Question 2: Changing the Strategy of the Users .	174
9.3.2.1	Changing The Strategy for User 1	176
9.3.2.2	Changing The Strategy for User 2	178
9.3.3	Addressing Question 3- Comparing the Results Between Users for Each Strategy	179

9.3.4	Addressing Question 4- Comparing the Results for Each User	181
9.3.5	Addressing Question 5- Comparing All of the Results	182
9.4	Second Scenario- Effects of Changes to the Environment	184
9.5	Third Scenario- Effects of Changes to the System Design	186
9.6	Fourth Scenario- Effects of Foreseeable Workload Changes	192
9.7	Conclusions and Other Uses	195
10	Discussion, Conclusions and Future Work	197
10.1	Overview	197
10.2	Issues in Using the Methodology and Modelling Approach Commercially	197
10.2.1	Issues Related to Data Collection	198
10.2.2	Issues Related to Building the Models	200
10.2.3	Issues Related to Generating the Results	200
10.2.4	Potential Extensions to the Modelling Approach	201
10.3	Conclusions, Research Contributions	203
10.4	Recommendations for Policy Makers	204
	Appendices	207
A	Information Sheets	209
A.1	Provided to NHS 24 COs	209
A.2	Provided to Reparticipating NHS 24 COs	211
A.3	Provided to Care Providers (System Version 1.2)	212
B	Consent Form	215
C	Interview Schedules	217
C.1	Questions for NHS 24 COs	217
C.2	New Questions for Version 1.2 for NHS 24 COs	221
C.3	Questions for Care Providers	223
D	List of Themes and Subthemes	227
D.1	For the First Part of the Study	227
D.2	For the Second Part of the Study	230
E	Detailed Description of the Improved Modelling Approach	233
E.0.1	Notions for the User Model	233

E.0.1.1	The Perceptual Buffer	233
E.0.1.2	The Belief Memory	237
E.0.1.3	The Conceptual Memory	238
E.0.1.4	The Skill Memory	241
E.0.1.4.1	Device Skills	242
E.0.1.4.2	External Skills	247
E.0.1.5	The Intention Memory	249
E.0.1.6	Goals	251
E.0.2	Notions for the Device Model	252
E.0.2.1	The Device State	252
E.0.2.2	The Device Actions	254
E.0.3	Behaviour of the User Model	257
E.0.3.1	Perception	257
E.0.3.2	Conceptual Inference	258
E.0.3.3	Skill Execution	261
E.0.3.4	Mechanisms for Remembrance/Forgetfulness	266
E.0.3.4.1	The Main Mechanisms	270
E.0.3.4.2	The Mechanism for Percepts to Remember	273
E.0.3.4.3	The (Infinite-time) Remembrance of In- formation as an Effect to an External Skill	275
E.0.3.4.4	Other Additions	276
E.0.4	Behaviour of the Device Model	277
E.0.5	Behaviour of the Controller	277
F	Detailed Requirements for Model Inputs in My Modelling Approach	279
F.1	Inputs for the User Model	279
F.2	Inputs for the Device Model	280
G	List of Settings	281
H	Glossary	283
	Bibliography	285

List of Figures

1.1	The use of telemonitoring systems in Telescot trials in Lothian ([21]) .	5
2.1	The Structure of Soar 9, taken from [133]	39
2.2	The Structure of ACT-R 5.0, taken from [35]	43
2.3	Process of building a simulation model, taken from [12]	51
3.1	The homepage	58
3.2	The Monitor page	59
3.3	The badge	59
3.4	The Alerts page	60
3.5	The Patient Monitoring page	60
3.6	The Surveys page	61
3.7	The first page of the Excel spreadsheet	62
3.8	The second page of the Excel spreadsheet	63
4.1	The Methodology	111
5.1	Mode confusion problems as detected by the controller	126
6.1	The structure of the new modelling approach	132
7.1	The Class Diagram	149
7.2	Process for Performing Repeated Runs	152
9.1	Total time distribution for user 1, strategy 1	175
9.2	Total time distribution for user 2, strategy 2	175
9.3	Total time distribution for user 1, strategy 2	177
9.4	Total time distribution for user 1, strategy 3	178
9.5	Total time distribution for user 2, strategy 1	179
9.6	Total time distribution for user 2, strategy 3	180

9.7	Differences between the time distributions for the three strategies for user 1	182
9.8	Differences between the time distributions for the three strategies for user 2	183
9.9	Total time distribution for user 1, strategy 2, large screen	185
9.10	Differences between the time distributions for user 1, strategy 2 for standard and large screen	186
9.11	Total time distribution for user 2, strategy 2, large screen	187
9.12	Differences between the time distributions for user 2, strategy 2 for standard and large screen	188
9.13	Total time distribution for user 1, strategy 2, all alerts on one page . .	189
9.14	Differences between the time distributions for user 1, strategy 2 for initial version and new design	190
9.15	Total time distribution for user 2, strategy 2, all alerts on one page . .	191
9.16	Differences between the time distributions for user 2, strategy 2 for initial version and new design	191
9.17	Total time distribution for user 1, strategy 2, in the case of an epidemic	194
9.18	Total time distribution for user 2, strategy 2, in the case of an epidemic	194
E.1	Percept components	234
E.2	Concept components	238
E.3	Device and external skill components	242
E.4	Components of intentions instantiating device skills and of those instantiating external skills	250
E.5	Device state components	253
E.6	Device action components	255

List of Tables

3.1	Number of recruited participants per team	65
7.1	Inputs required for the user model	150
7.2	Inputs required for the device model	151
8.1	Time taken for device actions/external skills for user 1	166
8.2	Time taken for device actions/external skills for user 2	167
9.1	Increase in the number of patients highlighted in each colour, having each combination of alerts	193

Chapter 1

Introduction

1.1 Current Challenges in Healthcare

The world's population is ageing, due to factors such as increasing longevity, decreasing fertility and the ageing of 'baby boomers' [48]. In Scotland, it was estimated that the number of people aged over 75 will increase by 82% between 2010 and 2035, from 0.41 million to 0.74 million [13]. Long-term conditions being prevalent in the older age, this will lead to growing numbers of adults suffering from, sometimes multiple, long-term conditions [118]. The latest Scottish Health Survey (from 2012) already showed that the number of adults suffering from long-term conditions in Scotland increased from 41% to 46% between 2008 and 2012 [52].

Also called chronic diseases, long-term conditions are characterised by durations of at least one year, negative effects on the patient's quality of life, and potentially continuous demand for care and support [197]. They can be either mental or physical, examples for the latter being epilepsy, inflammatory bowel disease, chronic pain, arthritis, asthma, diabetes, heart disease and chronic obstructive pulmonary disease (COPD). Long-term conditions already impose important financial and managerial burdens on the healthcare system. In Scotland, the latest Scottish Government Action Plan showed that long-term conditions account for 80% of all GP (general practitioner) consultations, over 60% of all bed days used, and are the cause of 60% of all deaths [197]. Moreover, they are twice as likely to lead to hospitalisations and much longer hospital stays.

The increasing demand for the care of people suffering from such conditions therefore constitutes a great challenge for the healthcare system. This is compounded by increased patient expectations of better care and equal access to healthcare facilities,

and a shortage of human resources in healthcare, reported by the European Commission ever since 2001 [9]. Due to demographic changes, it is expected that there will be fewer workers per dependant in general, and therefore even unpaid carers, whose life is already burdened by the care for family or friends, will be in shorter supply [118]. In this context, patient-centred models of management are severely challenged. Governments struggle to urgently find ways of reducing the future financial pressure in healthcare without affecting quality, equity and access [210]. As a result, new ways of organising and delivering health services are sought and experimented with.

1.2 A Potential Technological Solution

Telemedicine, one of the branches of eHealth, has been vigorously promoted by the European Commission for addressing the challenges faced by healthcare systems. The term “telemedicine” can be defined as *“the provision of healthcare services, through the use of ICT, in situations where the health professional and the patient (or two health professionals) are not in the same location. It involves secure transmission of medical data and information, through text, sound, images or other forms needed for the prevention, diagnosis, treatment and follow-up of patients”* [11, 16].

Telemedicine has many potential benefits for the management of long term conditions, as highlighted by recent European Commission Staff Working papers [11, 16, 102]:

1. **For patients**, telemedicine can improve health and the quality of life by reducing the need for face-to-face consultations (and the costs and anxiety resulting from travelling to them) and allowing for safe monitoring from home (telemonitoring), in a familiar environment.
2. **For the healthcare system**, telemedicine can improve the quality of healthcare and ensure a better management of the patients’ conditions by encouraging condition stabilisation and the avoidance of exacerbations, can reduce the number of human resources needed if the number of face-to-face consultations is reduced, can reduce costs by leading to fewer hospital admissions and shorter hospital stays.
3. **For the European economy**, telemedicine can make an important contribution. The 2012 European Commission Staff Working Paper points out that Europe has

a well developed industry in the areas of eHealth and telemedicine, including a high number of small and medium sized enterprises, and that this industry has seen a rapid growth and is continuing to grow [16]. They also estimate that the European market for eHealth has a potential value which represents one third of the global market for eHealth, and predicted that areas such as the telehome and the telehospital market in Europe could potentially bring more than double their financial value to the economy in 2016, as opposed to 2012.

1.3 Terminology Considerations

Despite still being used by the European Union, the term ‘telemedicine’ has been criticised in the past for being too exclusive, by implying specific uses and users [196]. In the UK, it has been progressively replaced by the term ‘**telehealth**’, which is used broadly to describe the use of technology for remote care either on a daily basis or in the case of acute scenarios. A detailed definition of telehealth was provided by the Scottish Government [118]:

“Telehealth is the provision of health services at a distance using a range of digital and mobile technologies. This includes the capture and relay of physiological measurements from the home/community for clinical review and early intervention, often in support of self management; and “teleconsultations” where technology such as email, telephone, telemetry, video conferencing, digital imaging, web and digital television are used to support consultations between professional to professional, clinicians and patients, or between groups of clinicians.”

As technology is moving towards the continuous monitoring of patients through the recording and transfer of data, the term ‘**telecare**’ has also been introduced [196]. Telecare is frequently meant to support elderly people suffering from a long term condition and who may be at risk, the main purpose being their safety. The Scottish Government defines telecare thus [118]:

“Telecare’ is the provision of care services at a distance using a range of analogue, digital and mobile technologies. These range from simple personal alarms, devices and sensors in the home, through to more complex technologies such as those which monitor daily activity patterns, home care activity, enable ‘safer walking’ in the community for people with cognitive impairments/physical frailties, detect falls and epilepsy seizures, facilitate medication prompting, and provide enhanced environmental safety.”

There is an increasing overlap between the uses and purposes of telehealth and

telecare, which has led to a new term gaining momentum in recent years for describing them together: **telehealthcare**.

I will use this more recent term throughout this thesis. However, as I am citing international papers, I will always mention if the authors have used different terms.

1.4 The Telehealthcare Services Under Consideration in This Thesis

This work was informed by experience with telemonitoring systems used as part of medical trials in Lothian, Scotland. ‘**Telemonitoring**’ is a term often used together with ‘telehealth’ to describe the remote monitoring of patients, who send in physiological data, by a care provider. It is the first use of telehealth which was mentioned in the definition of the term from the previous subsection. Telemonitoring is also applicable for telecare, as wearable sensors which can capture physiological data and send it to the care provider for remote monitoring are currently developed.

In Lothian, the medical trials carried out by the Telescot research team ([21]) have used telemonitoring both for patients suffering from potentially unstable long-term conditions (e.g. COPD), and for those suffering from relatively stable ones (e.g. diabetes, hypertension). Patients take their physiological readings (e.g. oxygen saturation and pulse for COPD) at home by using monitoring devices. Their interface to the system (e.g. that they have available on a tablet as part of the trial) may send them reminders to take their readings at certain times of the day, and may also require them to provide more information about their state by filling in a health-related survey. The readings and any survey results are sent via telecommunication links to a remote server, which presents them to the clinical or non-clinical monitors on a telemonitoring website. For potentially unstable conditions, call centre staff could be employed to monitor patients on this website daily, to eliminate any false alarms, request additional information from the patients if needed and urgently contact their nurse or care provider in case of exacerbations. For relatively stable conditions, practice nurses monitor patients on the system with less frequency (e.g. weekly). They may also contact the patient directly and, in case of worrying results, arrange an appointment for her with her GP. The GP may also use the website as a basis for decisions on changing the patient’s treatment, sometimes even during her consultation. A graphical representation of the uses of the telemonitoring systems in Telescot trials is presented in Fig. 1.1.

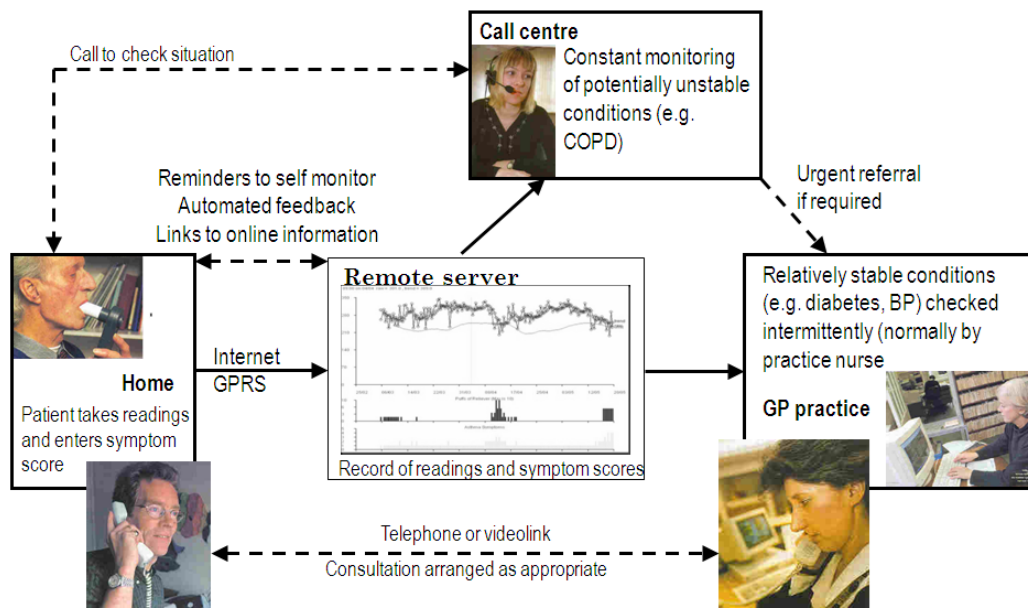


Figure 1.1: The use of telemonitoring systems in Telescot trials in Lothian ([21])

The telehealthcare ‘systems’ which I will use as examples throughout this thesis are the back-ends to telemonitoring systems (i.e. the telemonitoring websites and any related tools) used by clinical or non-clinical monitors for remote patient monitoring. Whenever I refer to the ‘user’, I will therefore not refer to the patient, but to the clinical or non-clinical monitors. Please note that, despite the examples being built around telemonitoring systems, the methods presented in this thesis are generic enough to be used with other telehealthcare systems and technology. An example of a telemonitoring system is provided in Chapter 3.

1.5 Is Telehealthcare Feasible?

Despite its potential advantages for patients suffering from long-term conditions, health-care systems and the economy ([11, 16]), the benefits of telehealthcare in terms of patient care and costs have not been convincingly proven yet.

A systematic review of reviews on the effectiveness of telehealthcare performed by Ekeland et al. (the authors refer to ‘telemedicine’ and ‘telecare’) concluded that, while some studies find telehealthcare as effective or promising, in others the evidence for the effectiveness of telehealthcare is unconvincing and incomplete [80]. In particular, in terms of effectiveness of care, there is not enough evidence of the benefits of telemonitoring for heart failure ([66]), diabetes management ([195]) and asthma care ([83]), and

a lack of evidence of the benefits of teleconsultation and video-conferencing for diabetes care ([212]) [80]. Some reviews underline that the lack of evidence is not a proof against effectiveness ([39]), while others advise caution in basing conclusions on narrow heterogeneous studies ([89]) or advise further research using a broader or different perspective, considering that not all of the important questions have been asked, at least not in the right way ([97]) [80]. With respect to the cost-effectiveness of telehealthcare, Ekeland et al. found that there is a general lack of knowledge, understanding and firm conclusions on this topic [80]. They only identified one review with positive findings on the cost effectiveness of telehomecare in terms of hospitalisations, patient compliance and quality of life ([193]). Other reviews were more cautious, pointing out the rarity and heterogeneity of the studies ([88]), their lack of quality ([209]), or the fact that effects would only become visible in the long term (on telemonitoring reducing travel time and hospital admissions for heart failure patients [200]) [80]. Yet other reviews were not conclusive about the cost-effectiveness of home telecare for older people and people who are suffering from long term conditions ([39]), synchronous telehealth in primary care ([77]), home respiratory monitoring ([111]) and diabetes care ([112]) [80].

Within a systematic review of reviews on the methodologies used in evaluating telehealthcare (the authors refer to ‘telemedicine’ and ‘telecare’), Ekeland et al. found that there is a general concern about the quality of the telehealthcare studies, their underlying methodologies and the questions that they can answer [77, 76, 93, 75, 98, 192, 216, 96, 81]. In particular, they point to reviews which criticise the lack of acceptable standards for economic evaluations ([216]), the potential lack of quality in the study design ([93, 98, 96, 74]) or in the use of qualitative evaluation methods ([161]), and the minimal consideration of the socio-economic effects of telehealthcare ([117]).

One of the reasons why telehealthcare has not proven as beneficial yet is the fact that the evidence supporting it often comes from small scale and heterogeneous studies, which makes it difficult to draw general conclusions [154]. In the UK, the NHS (National Health System) was blamed for having “more pilots than British Airways”, but which often do not share their findings and experience, and are sometimes not even aware of each other [69, 213]. Moreover, the pilot studies were criticised as being purely academic and taking too long, with results being published when technological and cultural problems will have already been solved, which gives an opportunity to sceptics to criticise and reject the technology [213]. Some critics labelled the evidence of small scale trials as ‘equivocal’ ([135]), and positively influenced by enthusiasts or technology

manufacturers who are careful in the selection of research sites and participants ([153]).

One criticism of small scale pilot studies in the UK which is particularly important for this work is that the management of patients by using the telehealthcare systems is often added to current, no longer effective, clinical pathways and for a small percentage of the patient population. The scaling up of the telehealthcare deployments would require the complete redesign of clinical pathways to include the use of the technology for all the patients it is intended for [72, 101]. In the current setting, if pilots fail at small scale, they will have temporarily added an extra layer on clinical pathways in the sites where they were carried out [72]. If they are successful at small scale, it is not clear whether the benefits of the telehealthcare system deployments would be achieved when they are scaled up. The general conclusion is that the deployment of telehealthcare should be joined by, and help inform, the redesign of care pathways.

1.6 European Projects Scaling up Telehealthcare

Motivated by the unconvincing results on the effectiveness of telehealthcare at small scale, several European countries have been planning and developing large-scale telehealthcare pilots, which will help to more realistically and thoroughly assess whether telehealthcare would work at scale. Two important examples are the **Renewing Health European project** ([7]) and **ITTS (Implementing Transnational Telemedicine Solutions)** ([1]). **Renewing Health** involves nine European regions in the provision of telehealth services for large segments of the population suffering from chronic obstructive pulmonary disease (COPD), diabetes and cardiovascular disease. The **ITTS project**, partly funded by the EU Northern Periphery Programme, involves partners from six Northern European countries, and aims to facilitate transnational knowledge exchange for implementing telehealthcare solutions at scale.

Starting in 2008, the UK carried out the **Whole System Demonstrator** programme ([219]), involving 3230 patients suffering from COPD, heart failure or diabetes in a 12-month trial period, across three sites in England: Newham, Kent and Cornwall [204]. Its purpose was that of assessing the effects that a scaled up delivery of telehealthcare technologies would have, as compared to traditional care, on secondary care and mortality, the patients' quality of life and cost effectiveness. It was also intended to shed more light on the implications of such a scaled up delivery for patients, professionals and healthcare organisations. Despite some excellent preliminary results, in which the trial claimed that the use of telehealth leads to reduced mortality, hospital admissions

and hospital stays ([219]), its final results were not very convincing. In terms of hospital admissions, it was found that fewer patients using telehealth than control patients using traditional care were admitted to hospital. However, this difference was relatively small and potentially impacted by a large increase in emergency admissions by the control group at the beginning of the trial [204]. Differences between patients using telehealth and the ones using traditional care in terms of emergency department visits, elective admissions, outpatient attendances and tariff costs were found insignificant. Telecare also did not lead to significant reductions in health and social care service use [205]. In terms of mortality, it was found that many fewer patients using telehealth died as compared to control patients (4.6% as opposed to 8.3%), but critics wonder in what measure telehealth contributed to this difference ([204, 60]). With respect to telecare, it was found that it does not lead to significant differences in mortality [205]. In terms of care, telehealth was not found to improve the quality of life or psychological state of the patients as compared with traditional care [64]. In terms of cost effectiveness, it was found that the telehealth intervention actually cost more than traditional care [95].

The UK is currently involved in another large-scale deployment of systems such as these through the **Delivering Assisted Living Lifestyles at Scale (DALLAS [23])** programme. DALLAS was developed jointly by the UK's Innovation Agency and the Technology Strategy Board (TSB), and has had a total investment of 37.3 million, out of which 25 million from the Scottish Government [23]. Its overall purpose is that of leading to the progress of telehealthcare assistive technologies from small-scale medical trials to large-scale deployments, which will enable access to them to millions of people across the country [15]. Proponents believe that this will help show how telehealthcare assistive living technologies can be used to improve health and wellbeing and encourage independent living. The 2011 Scottish Government document ([14]) places large-scale deployments at the centre of Scotland's health strategy. The plan is for DALLAS to involve, by 2015, almost 170,000 people across several areas in the UK, and to provide them with innovative systems, products and services which will help them self-care as they age [23]. The project consists of four consortia which will run and test it across different areas of the UK. The consortium from Scotland developed the 'Living it up' programme, which aims to promote independent living, health and wellbeing by connecting people, whether at home or not, using the familiar devices that are right for them, and encouraging the use of their 'Hidden talents' in their community [23, 2]. In November 2013, the project launched an online service which enables patients to get personalised health and wellbeing advice and be linked

to useful local services, events and activities in their community [17]. The service can be accessed through familiar technologies such as televisions, tablets or smartphones. 5,000 users were involved in its pilot study and current users are encouraged to help shape it further by proposing local services that they would like it to include.

In Scotland, the Telescot programme of academic research led by Prof. Brian McKinstry from the University of Edinburgh explores the effects of telemetric¹ supported self monitoring for a range of long term conditions [21]. The programme is a collaboration between private and voluntary organisations. One of its aims is that of unveiling, through both quantitative and qualitative methods, the effects of deploying telehealthcare in terms of early intervention and clinical outcomes, costs and user acceptance. Projects which are relevant for this thesis involve telehealthcare systems for home blood pressure monitoring during pregnancy ([19]), COPD ([181]), type 2 diabetes ([20]), congestive heart failure ([82]), hypertension ([155]), and nurse-led monitoring in stroke or transient ischaemic attack [21].

1.7 Thesis Claim

Even if deploying similar technology, small scale telehealthcare pilots nowadays can differ substantially both in terms of the contexts where they are implemented and the protocols governing their implementation. For telemonitoring systems **contexts may differ**, for example, in terms of:

- **The users:** Apart from personal differences in terms of capabilities, preferences, attitude towards technology, which always distinguish individuals, users can differ in terms of professional experience (with their work, with technology, with similar systems) and, depending on the healthcare organisation where they work, in terms of roles and familiarity with the patients. For example, healthcare professionals treating a small number of patients in small, local, health centres tend to develop a closer relationship with their patients and thus know better what treatments work for them and what generates anxiety. Also, depending on the size, location and speciality of healthcare organisations, only GPs can prescribe treatments to their patients, or both themselves and nurses can do that.

¹‘Telemetry’ is “The science and technology of automatic measurement and transmission of data by radio or other means from remote sources to receiving stations for recording and analysis” [5]. It is a term often used in conjunction with telehealth and telecare

- **The workload:** Workloads probably constitute the largest element of variation between telehealthcare deployments, as they are affected by a multitude of quantitative and qualitative, internal (to the healthcare organisation) and external aspects. For example, while they are given by the number of patients to monitor and how sick they are (which may also be affected by other health conditions, past medical history, lifestyle), they are also influenced by external factors such as seasons, the weather, or epidemics. Telemonitoring work involves the users calling patients to find out about their condition on the phone, ask them to repeat readings, and offer them advice, which all adds up considerable time. The ease of contacting the patients and the duration of the conversation is affected by their characteristics such as their age, preferences and behaviour, how anxious they are about their condition, how close a relationship they have with their care providers, etc. All of these elements influence workload.
- **The additional tools or systems used:** Currently there is no integrated technology available for the different aspects of patient care in the UK, and different health organisations use different systems for health records, telehealthcare, on-line prescriptions, getting lab results, etc.
- **The general organisation of the work and the clinical pathways used.**

The **choices made for protocols for telehealthcare pilots can also differ substantially** in terms of:

- **Roles and responsibility allocation:** For example, in some pilots patients are first checked on the system by non-clinical call operators from a call centre, who sort out false alarms and forward cases which need attention to nurses or GPs. This helps decrease the workload of care providers, but also requires the hiring of the call centre staff. In other pilots, clinical (nurses) or non-clinical monitors employed within the healthcare organisation sort out false alarms and forward cases which need attention to GPs. In this case, only the clinical monitors can also offer advice, and sometimes even make changes to the patient's treatment if necessary. Yet in other pilots, the whole monitoring work is performed by both nurses and GPs, and only GPs or both can change the patient's treatment.
- **The characteristics of the workload:** Pilots normally include strict requirements for the profile of the patients to be recruited for being monitored by using the system, and this is not consistent between them.

- **The organisation of the work involving the use of the system and how this is included in existing clinical pathways:** Protocols can include requirements for the users to work in a team or by themselves, constrain the frequency and duration of this work, and how it should be performed. This also varies between pilots.
- **The type of information and level of support offered by the system.**
- **The help and support offered by the protocol** (e.g. who is on site to help the system's users with technical or care issues, who they can contact, the paper documentation coming with the protocol, etc.).

This great heterogeneity makes it difficult to predict the success of a telehealthcare deployment, and what would need to be changed in terms of the protocol and clinical pathways, to encourage success in new contexts. Given the current interest of European countries to scale up telehealthcare deployments, and the associated risks, such a prediction becomes desirable in order to motivate investment in each new context in the process of scaling up, and help save resources. It is this problem that I would like to tackle in this thesis.

If, once a deployment is found to be successful in one context, we explore the differences brought by a potential context in terms of the users, workload and ways that the system would be used as part of the clinical pathway, this can help us predict the deployment's success there. If the findings predict success, this motivates the scaling up of the telehealthcare system deployment to incorporate the new context. If, nevertheless, the findings are not encouraging, we can investigate 'what-if' changes to the potential context in terms of the users, workload, ways that the system is used as part of the clinical pathway, or the system design itself, and predict the deployment's success given each. This could suggest the changes that would be necessary for ensuring success there. 'What-if' scenarios are also useful for investigating the effects on success of hypothetical changes to one context over time, and help mitigate risks. For example, would the telehealthcare service still be successful if there is an increasing workload due to patients triggering more alerts during a flu epidemic, or in the case of decreasing numbers of users?

Issues such as increased productivity through the more timely high quality remote care of increasing numbers of patients, reduced error rates to ensure patient safety, and an interface and functionality which fit the users' needs to increase satisfaction and encourage acceptance, are all important for today's telehealthcare systems. Such issues

being central to the notion of usability, this makes the usability of telehealthcare systems a very important factor for their success.

Efficiency is seen by many authors as a characteristic of usability, but it is also influenced by other usability characteristics. We define efficiency as “the probability distribution of the time spent by a healthcare worker in monitoring patients, by following a certain work process, on a telehealthcare system”. Efficiency is an important factor for success, due to the predicted shortage of human resources in healthcare, on one hand, and the increase in their workload due to an increase in the number of patients suffering from long term conditions, on the other [11, 18].

The efficiency of a telehealthcare system can be investigated before the system is deployed by performing a usability evaluation. However, as was found for many software companies in general, pre-deployment usability evaluations are often very expensive from a financial and time perspective. This often prevents companies from properly performing them, leading to software which does not completely meet its usability requirements [87]. As observed during my Masters for telehealthcare systems, this leads to usability problems, including inefficiency, which may be hard to foresee in general, and even more so if the system is to be deployed in heterogeneous contexts [25]. Moreover, it has been noted that usability has been only considered at a very late stage in the history of medical informatics ([206]), and that it is seen as secondary in the development of telehealthcare solutions ([202]).

The efficiency of the work process surrounding the telehealthcare system is influenced by the usability (and efficiency) of the system in general. In particular, for example, the number of times a care provider managing patients by using a telemonitoring system needs to telephone a patient to find out more details about her state, and the duration of the calls, will be influenced by the fitness for purpose of the system’s algorithms alerting her as to the patient’s state, and the clarity in which the information on the patient’s state is presented. In the absence of a proper pre-deployment usability evaluation, it is therefore also hard to predict the efficiency of the work process surrounding the telehealthcare implementation.

To address these issues, I will investigate how efficiency can be predicted in potential, real or hypothetical, contexts for telehealthcare system deployments, but at the same time present approaches which are broad enough to also deal with other aspects of usability for such contexts. As it is often the case that actions performed in their environment by the users of a telemonitoring system as part of their monitoring work (e.g. telephoning patients, consulting medical records, etc.) are much more time

consuming than those performed on the system itself, I will consider not only the efficiency of the system, but also that of the work process surrounding its use.

The work presented in this thesis is an experimental investigation, but my vision is that, if used commercially, it could help reduce risks in projects scaling up telehealthcare. I should add that, of course, interest in predicting the success and, in particular, efficiency as we define the term, are generally desirable for any software system. Having said that, the techniques that will be described throughout this thesis are general enough to be implemented for any software system with an interface, but were developed specifically for deployments of telehealthcare systems due to the important motivation for them in this domain.

1.8 Aims and Objectives

Given all of the above considerations, the main aim of this thesis is to address the following question:

How could we use modelling and simulation techniques, and lessons learned about the efficiency of a telehealthcare system and work process within a reference deployment, for predicting the same types of information in potential deployments of the system? These potential deployments could be real ones, where the focus is on rolling out the system, as well as hypothetical ones, resulting from ‘what-if’ experimentation with changes to an existing deployment.

This main aim can be broken down into the following objectives:

1. To propose a **structured methodology** which can guide a modeller for systematically obtaining predictions regarding certain usability characteristics for a telehealthcare system and work process surrounding it in potential, real or hypothetical, deployments of the system, as departing from lessons learned about the same usability characteristics from one reference deployment. The methodology is intended to be generic by working with any type of modelling approach.
2. To develop an appropriate **modelling approach for building user and system models** which can be used together with the methodology for predicting the efficiency of the telehealthcare system and work process within one deployment.
3. **To collect input data and qualitative observations from a real telehealthcare system deployment** for building user and system models by using the modelling approach.

4. **To instantiate models by using the collected inputs, where available, or by making assumptions based on qualitative observations, where not, and running a series of fictional, yet realistic, examples.** These examples are meant to showcase how my methodology and modelling approach could be used to predict the efficiency of a telehealthcare system and work process in hypothetical deployments.
5. **To draw conclusions about the benefits of using the methodology and modelling approach, in the light of the obtained results.**

1.9 Summary of the Thesis

The objectives of this thesis are addressed in turn within the following chapters.

Chapter 2 provides an outline of related literature from the field of Human Computer Interaction, but also other fields which are concerned with the prediction of efficiency in a work context. I discuss the advantages and disadvantages of different approaches for usability evaluation and explain my choice of the approach presented in this thesis.

Chapter 3 is especially relevant to readers who are interested in telehealthcare. It describes the methodology and results of a study that I have performed in Lothian, Scotland, for evaluating a telehealthcare system used by call operators and care providers. The results motivate the focus of this thesis, that of developing an approach for automatically predicting the efficiency of the system and work process surrounding its use.

Chapter 4 describes the methodology that I have developed for helping to reuse models to explore characteristics of usability in different, real or potential, telehealthcare system deployments. It also highlights the main advantages brought by this methodology.

Chapter 5 is especially relevant to readers who are interested in usability evaluation, and my exploration of approaches for it. The chapter describes a first modelling approach that I have developed to be used alongside the methodology, for the purpose of predicting problems which cause mode confusion. I found that this approach had some important disadvantages, which are also described.

Chapter 6 presents an improved modelling approach, and the most important contribution of this thesis. It includes a high level description of the structure, behaviour

and uses of the modelling approach, such that **it could be interesting to all of the readers. Readers who are interested in usability evaluation methods are referred to Appendix E for a more detailed description of the modelling approach.**

Chapter 7 provides an overview of the Java application which I have developed for implementing the modelling approach, its structure in terms of classes, and the way that it can be used. **I have also kept this chapter as high level such that it could be interesting to all of the readers.** For more details about how to instantiate the models, readers are referred to Appendix F.

Chapter 8 returns to the usability evaluation study which was described in Chapter 3, to explain how I have collected data for my approach from it, and the way that this data served to instantiate my initial models.

Chapter 9 describes the methodology and modelling approach in action. I show how they can be used to explore efficiency in the case of different user levels of experience, ways of using the system, layouts of the workplace, system designs, and, most importantly, changing user workloads in the case of an epidemic.

Chapter 10 provides a critical discussion of the issues which should be considered for commercialising the work presented in this thesis, and sums up the contributions of the thesis. It ends with some recommendations for making Scotland a friendlier environment for carrying out health IT research, as drawn from my experience.

Chapter 2

Literature Review

2.1 Overview

In this chapter I will review the most relevant literature for this thesis. I will start by explaining the notion of usability and its general advantages, as well as those specific to medical designs. I will then explain my decision regarding a definition for efficiency to be used in this thesis, in the context of telehealthcare system deployments. I will review the different methods used for evaluating usability, and explain the advantages and disadvantages of each. The techniques that I have decided to use in this thesis are model-based evaluation methods based on cognitive architectures. I will explain why I have not chosen other model-based evaluation methods, such as the GOMS (Goals, Operators, Methods and Selection Rules [62]) family of techniques.

My work uses modelling and simulation to avoid the costly involvement of users, so I will provide a brief overview of this field and the advantages of using its methods. As the estimation of time for planning purposes and improving work efficiency have long been central subjects in the management literature, I will also review some techniques used for predicting efficiency that were developed in this field.

2.2 The Issue of Usability

2.2.1 Definitions and Characteristics of Usability

The term “**usability**” was initially derived from that of “**user friendliness**”, defined as “an expression used to describe computer systems which are designed to be simple to use by untrained users, by means of self-explanatory or self-evident interaction

between user and computer” ([65]) [30]. However, the term “user friendliness” was later considered as having become too vague and subjective ([43]), or incomplete ([170]) [30]. This led to the coining of different definitions of usability by different standardisation organisations and experts, and no consensus as to a widely accepted and precise one, or as to the factors affecting usability [24, 30]. I will present here the most cited definitions.

Among the definitions proposed by standardisation organisations, the one provided by the International Organisation for Standardisation in the **ISO 9241-11 standard** is the best known: “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [107]. The standard provides rather abstract definitions of the three characteristics of usability [211]:

- **Efficiency:** “resources expended in relation to the accuracy and completeness with which users achieve goals” [107]
- **Effectiveness:** “accuracy and completeness with which users achieve specified tasks” [107]
- **Satisfaction:** “freedom from discomfort and positive attitudes towards the use of the product” [107]

The Annex B of the ISO 9241-11 standard further explains what metrics should be considered for assessing the three characteristics: “percentage of goals achieved, percentage of users successfully completing tasks and average accuracy of completed tasks” for effectiveness, “time to complete a task, tasks completed per unit time and monetary cost of performing the task” for efficiency, and “rating scale for satisfaction, frequency of discretionary use and frequency of complaints” for satisfaction [107]. Several authors have criticised the abstractness and lack of practicality of the standard [211, 199].

Another well-known definition of usability was provided by standard **ISO/IEC 9126-1 (2001)**: “the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions” [108]. However, in the latest update to the standard, **ISO/IEC 25010:2011**, a decision was taken to adopt the definition of the ISO 9241-11 standard instead in order to ensure consistency [109]. In this new standard, usability is seen in its broad sense as a subset of “quality in use” for products or systems, with its three characteristics: effectiveness, efficiency

and satisfaction. The notion of “**quality in use**” is used to represent the user’s overall experience with the system’s quality. The standard also presents usability as a general component of **product quality**, with another six characteristics [109]:

- **Appropriateness Recognisability:** “Degree to which users can recognize whether a product or system is appropriate for their needs”
- **Learnability:** “Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use”
- **Operability:** “Degree to which a product or system has attributes that make it easy to operate and control”
- **User Error Protection:** “Degree to which a system protects users against making errors”
- **User Interface Aesthetics:** “Degree to which a user interface enables pleasing and satisfying interaction for the user”
- **Accessibility:** “Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use”

Among the numerous experts who have provided definitions of usability, the definition given by **Jacob Nielsen**, who is seen as the Father of Usability, is probably the most cited: “Usability is a quality attribute that assesses how easy user interfaces are to use” [173]. Nielsen further decomposes usability into 5 quality components [173]:

- **Learnability:** “How easy is it for users to accomplish basic tasks the first time they encounter the design?”
- **Efficiency:** “Once users have learned the design, how quickly can they perform tasks?”
- **Memorability:** “When users return to the design after a period of not using it, how easily can they re-establish proficiency?”
- **Errors:** “How many errors do users make, how severe are these errors, and how easily can they recover from the errors?”

- **Satisfaction:** “How pleasant is it to use the design?”

Preece et al. define usability as “a measure of the ease with which a system can be learned or used, its safety, effectiveness and efficiency, and the attitude of its users towards it” [186]. In more recent work, the authors propose the breaking of the term into a series of **usability goals** [187]:

- **Effectiveness:** “Is the system capable of allowing people to learn well, carry out their work efficiently, access the information they need, buy the goods they want, and so on?”
- **Efficiency:** “Once users have learned how to use a system to carry out their tasks, can they sustain a high level of productivity?”
- **Safety:** “Does the system prevent users from making serious errors and, if they do make an error, does it permit them to recover easily?”
- **Utility:** “Does the system provide an appropriate set of functions that enable users to carry out all their tasks in the way they want to do them?”
- **Learnability:** “How easy is it and how long does it take (i) to get started using a system to perform core tasks and (ii) to learn the range of operations to perform a wider set of tasks?”
- **Memorability:** “What kinds of interface support have been provided to help users remember how to carry out tasks, especially for systems and operations that are used infrequently?”

The usability goals are seen as separate from **user experience goals**, which include what other authors defined as satisfaction for usability, as well as elements of fun, motivation and helpfulness [187].

Alonso-Rios et al. criticized the briefness and lack of clarity of the different definitions, and the limited number of, usually informal, examples that support them, and proposed what I have identified as the most complete taxonomy of usability characteristics and sub-characteristics [30]. They propose the following characteristics and first level of sub-characteristics [30]:

- **Knowability:** “the property by means of which the user can understand, learn, and remember how to use the system”. It is broken down into clarity, consistency, memorability and helpfulness.

- **Operability:** “the capacity of the system to provide users with the necessary functionalities and to permit users with different needs to adapt and use the system”. It is broken down into completeness, precision, universality and flexibility.
- **Efficiency:** “the capacity of the system to produce appropriate results in return for the resources that are invested”. Its sub-characteristics are efficiency in human effort, efficiency in task execution time, efficiency in tied up resources, efficiency in economic costs.
- **Robustness:** “the capacity of the system to resist error and adverse situations”. Its sub-characteristics are: robustness to internal error, robustness to improper use, robustness to third party abuse and robustness to environment problems.
- **Safety:** “the capacity to avoid risk and damage derived from the use of the system”. Its sub-characteristics are: user safety, third party safety and environment safety.
- **Subjective satisfaction:** “the capacity of the system to produce feelings of pleasure and interest in users”. It is broken down into interest and aesthetics.

2.2.2 Benefits of Usability

Despite a lack of consensus regarding a unified definition of usability and its characteristics, authors agree on the benefits of good usability. As described by Maguire, a system with good usability has been shown to [149]:

- **Increase user productivity**, as the system would be built according to usability principles and with the user’s needs and tasks in mind.
- **Reduce the need for training and support**, as learning would be facilitated by the interface and users could learn to use the system just by trying it out.
- **Reduce error rates** by eliminating the high proportion of errors that users make because of inconsistencies, lack of intuitiveness or other design issues.
- **Increase the user acceptance of the system**, as users prefer a well-designed system which caters for their needs, is easy to use and presents information in a clear, accessible way.

- **Enhance the reputation of companies producing such usable systems**, because of the enhanced acceptance and high regard of the system's users and customers.

All of the advantages from above make usability very important for telehealthcare systems from the perspective of the clinicians and non-clinical monitors. In particular, these systems are intended to help with the management of an increasing number of patients, by decreasing numbers of healthcare staff, and therefore support for good user productivity is important. Errors can be vital to patients in safety critical systems such as telehealthcare systems, and thus their avoidance is essential. The introduction of telehealthcare brings about important changes to healthcare organisations. As highlighted by Hjelm, one difficulty is that some healthcare workers are reluctant to change their work practices for accommodating it (here using the term 'telemedicine') because ([102]):

1. There is not enough proof yet regarding its efficacy and cost-effectiveness.
2. It may threaten their role and status.
3. It will increase their workload.
4. It does not cater for their knowledge and skills.
5. Off-the shelf technology will not cater for cultural and language differences.
6. There is a lack of standards.

This therefore makes acceptance an important factor. Karsh explains that a telehealthcare system (using the term 'telemedicine') with good usability- which is useful, easy and pleasurable to use- will encourage healthcare workers to overcome the anxiety caused by its introduction in their healthcare organisation [124].

Broens at al. conducted a literature review of conference papers describing telehealthcare interventions (here the authors using the term 'telemedicine') [51]. They concluded that the determinants of good telehealthcare implementations can be classified into five categories: those related with the telehealthcare technology itself, the acceptance of telehealthcare, financing, the healthcare organisations and policy and legislation, out of which technology and acceptance were the most frequently reported. The authors point out that usability is very important, being both a technical and an acceptance determinant, as it influences the system's usefulness and also encourages

acceptance. They underline the fact that telehealthcare systems are not always built with their users, patients or healthcare workers, and the tasks that they undertake in mind, and that more attention should be paid to ensuring comfortable use for patients, and clarity of information and flexibility of services for healthcare workers.

2.3 Considerations of Efficiency in This Thesis

As we noticed in subsection 2.2.1, efficiency is a recurring theme in definitions of usability. Nevertheless, its own definition varies between them, and the meaning it has in one definition might span across other usability characteristics in another. In particular, while all of the definitions refer to the relationship between the resources spent and system outputs, there are inconsistencies between the different types of resources that they refer to (e.g. time, effort, task execution time, financial resources). Some definitions (those provided by Nielsen [173] and Preece [187]) only consider experienced users. Moreover, in some definitions efficiency seems to be related to learnability (but for novice users for Nielsen [173], or as an element of product quality for ISO/IEC 25010:2011 [109]), and operability (for Rios [30]).

I have therefore decided on my own definition of efficiency, in the context of telehealthcare systems, which was already mentioned in the first chapter of this thesis:

Efficiency= “the probability distribution of the total time spent by a healthcare worker in monitoring patients, by following a certain work process, on a telehealthcare system.”

As Nielsen ([173]), I am therefore describing efficiency in terms of *time*, more exactly *task execution time*. The time required by both novice and experienced users (clinicians or non-clinical monitors) to monitor patients on a telemonitoring system is important for a healthcare organisation in its analysis of time allocation and personnel needs. Unlike Nielsen, I am therefore considering for my definition users with all levels of experience.

As I observed in my previous work ([25]) and during a usability study which will be described in Chapter 3, a large proportion of the time spent in doing monitoring work is due to steps performed by the user in her environment, and not on the telemonitoring system itself. For example, it takes users much longer to telephone patients than to check on their state from the system. Moreover, it is essential for usability to consider the user environment and task at hand. **I am therefore considering efficiency in terms**

of both the system and work process surrounding it.

While this thesis only considers one measure for efficiency- task execution time- some of the definitions which were presented in subsection 2.2.1 describe the term as multi-dimensional. In particular, apart from considering task execution time, the ISO 9241-11 standard ([107]) also uses as metrics the number of tasks completed per time unit and monetary cost, and Alonso-Rios et al. ([30]) also include as sub-characteristics efficiency in human effort, tied up resources and economic costs. To deal with such multi-dimensional definitions, I propose as future work the extension of the approaches presented in this thesis.

I also acknowledge the fact that efficiency is not everything. A very efficient system, which allows users to work very quickly, may nevertheless lead to increased error rates or a reduced quality of the work. Such negative effects may counterbalance the benefits of efficiency and, in the case of safety-critical systems such as telehealthcare systems, even endanger lives. Both error rates and the quality of the work, as well as other aspects which were mentioned by other definitions of usability (e.g. learnability, operability, accessibility, memorability, satisfaction), should be considered as other factors contributing to the success of a system in its deployment. Approaches like the ones proposed in this thesis, for the prediction of system and work process efficiency, should therefore be joined by an analysis of other measures to offer a full picture of the system's success.

2.4 Introduction to Usability (and Efficiency) Evaluation

The validation of usability requirements is named evaluation in HCI (Human Computer Interaction)[186, 78]. As for Software Engineering, the HCI community see evaluation as a process which should occur throughout the whole system life cycle, starting from the evaluation of prototypes or even paper mock-ups of the design and ending with the evaluation of the finalised product, feeding back into the design at every stage in between [78]. The goal of evaluation is to ensure that the system's functionality respects user requirements, to identify specific design problems, as well as to ensure that the system is easy to learn, usable and that users are satisfied with it. The evaluation may be formative, if it informs a new iteration of the product, or summative, if it aims to draw conclusions on the finished product [186].

Usability evaluation puts a greater emphasis than Software Engineering Verification and Validation on the importance of involving users, apart from experts, in the evaluation

of each iteration of a software product for feeding back problems for a new iteration [78]. This is explained by the difficulties that even experts in human cognition have in deciding on user cognitive demands for interacting with a design if users are not involved.

Within an idealised standard usability design process, once a first prototype is built following requirements, the designer's intuition and usability guidelines, a representative set of users evaluate the design to decide on existing usability problems [128]. The level of criticality of the identified problems could then motivate the revision of the prototype, and its new evaluation with users, within a new iteration. Future iterations could become necessary, as new usability problems could emerge after each iteration. This process ends when either no more critical problems are identified with the software system, or when there are not enough resources left for new iterations.

This usability design process is well documented and proven to work ([136]), and a considerable amount of knowledge and experience have been accumulated with regard to guidelines for good design and the evaluation with users [128]. Nevertheless, as we will see in the following sections, the involvement of users in the evaluation has some important limitations, which have motivated the use of alternative evaluation methods, involving usability experts or even automating usability evaluation.

No matter the usability evaluation method used, the main issues that need to be considered before performing the evaluation are the users' characteristics, the activities users perform or are predicted to perform, the environment in which the evaluation will take place and the development phase of the system to be evaluated [186]. User and task-related information is important even for predictive, expert-based evaluations, as experts need to base their predictions on knowledge about typical users.

2.5 Usability (and Efficiency) Evaluation Methods

For this section, I will mainly use the classification of usability methods proposed by Zhang ([222, 223]), adding to it model-based evaluation methods, as they were described by Kieras ([128]). I will group the methods into *empirical methods*, which involve users, and *analytical methods*, which are carried out by experts.

2.5.1 Empirical Methods

As a system should be built for its users, it is considered that no other evaluation method could replace the evaluation with real users [78]. User participation tends to take place during later development stages, when at least a simulation of the prototype is in place, or when a functional prototype or even the whole system is available.

2.5.1.1 Testing Methods

Usability testing is one of the most frequently used methods for evaluating the usability of a system throughout its lifecycle. It became important ever since the 1980s when software started being used by the general public [79]. It has become essential, as it provides concrete information about how users interact with the system and what their difficulties with it are [170].

Usability testing is carried out within an experimental setting and consists of the main steps involved in an experiment [104]. It involves having users work through a list of tasks on the system while being observed by the evaluator, with the purpose of discovering usability problems.

The following are some of the important usability evaluation methods used for testing:

- **The thinking-aloud protocol:** One of the most frequently used evaluation techniques, which involves encouraging the user to explain everything that she is thinking while performing tasks on the system ([170]). It helps the evaluator identify user misconceptions about the system's functionality, the parts of the system which cause problems and the things that users do or do not like. The method requires little expertise and is very useful for gathering qualitative data from a fairly small number of users, but it cannot be used very well for gathering quantitative data. Another important disadvantage is that it can distract the evaluator from finding real problems if she is too concerned about the users' opinions regarding these problems. Also, it is unnatural for most users and can impact the results due to some users' difficulty to express what they are thinking while carrying out tasks.
- **The question-asking protocol:** More natural than the thinking-aloud protocol, it also involves the user verbalising her thoughts while performing tasks on the system or prototype, but also the tester asking her direct questions [222].

This allows the tester to understand the user's mental model, her difficulties and insights.

- **Cooperative evaluation:** A variation of the think-aloud protocol which encourages the user to see herself as a collaborator who will answer, as well as ask questions and get responses from the evaluator [78]. To decide on the reasons behind actions that users perform during observations, a post-task walkthrough can be carried out by showing or playing to the user the collected data and asking her to comment or state the reasons for her actions. This method is especially suitable for getting explanations about critical or intensive tasks which the user could not verbalise during the think-aloud.
- **The shadowing method:** Is performed when the user cannot think aloud, and involves an expert user sitting next to the tester to explain the user's actions on the system [222].
- **The coaching method:** Involves the user asking the tester, or an expert user with the tester observing, questions about the system while performing the predefined tasks [170, 222]. It helps the tester understand user needs in terms of training and documentation, and how the system could be improved for her. A skilled tester can also adapt her answers during a series of tests to find what helps the user the most.
- **The teaching method:** A test user gets acquainted with the system first, and then explains to a novice user how to perform a set of predefined tasks [214, 222]. The novice user is debriefed about not doing any problem-solving during the demonstration, but allowing it to be done only by the test user. The method helps assess the learnability of the system.
- **Co-discovery learning:** Two users are asked to help each other in performing tasks on the system, as if working together, and verbalise their thoughts, while the tester observes their interaction [170, 222].
- **Performance measurement (benchmarking):** Normally involves having a group of test users perform some test tasks within a certain timeframe on the system, and recording quantitative measures of user performance for certain usability goals [170]. *Usability specifications* are used for guiding the benchmark tasks so that they stop at the system iteration in which user performance falls within

required limits to indicate good usability [186, 78]. These specifications contain within a tabular form the usability attribute to be inspected, the way the data is to be gathered and limits (worst case, best case, planned, actual) for quantifiable measures of the usability attribute (in terms of time, number of errors, number of times users needed to consult the documentation, etc.) which would make it acceptable for ensuring usability. User performance can be tested through observation either in a laboratory built for the purpose, where users are required to perform a set of given tasks, or within their work environment, where observation of the same tasks will be of interest [170]. The users' opinions are important, therefore debriefing is also carried out with users through interviews or questionnaires and the obtained data is recorded within the usability specification in the form of attitude metrics [186].

- **Remote Testing:** Is used for tests in which the tester and the user are in different places, or data collection and analysis take place at different times [94, 222]. Depending on space and time, there can be *same-time-different place testing (remote-control testing)*, in which the tester observes, and sometimes even hears the user remotely through a network, or *different-time-different-place testing*, such as the *journaled session* described by Nielsen ([170]), in which user actions as part of the test session are guided and recorded for later analysis.
- **Retrospective testing:** Can be used after performing any test method in which the user and tester cannot interact, if the test session was videorecorded [170, 222]. The tester watches the videotaped session together with the user and asks her questions about her behaviour and reactions during the test. This method can help collect valuable data, but it takes at least twice as long as the test session, so it is very expensive.

2.5.1.2 Inquiry Methods

Inquiry methods also involve users, but their purpose is to collect qualitative data on opinions, comments and suggestions [110]. They can be used during usability testing, for needs assessment during the early stages of the system's design or for collecting user opinions for the post-release improvement of the interface.

The following are the most important types of inquiry methods:

- **Ethnographic studies/ field observation:** Involves observing the users in the

field, in the real context of use of the system, as opposed to the artificial context of a laboratory for usability testing [170, 104]. The evaluator spends a few hours in the user's workplace to collect observations and to interview users about their work and their use of the system. This type of evaluation is best performed in the early stages of system development, to elucidate user needs and requirements.

- **Contextual inquiry:** A structured field interviewing method employed over a long-term study, useful for discovering work practices, understanding the work context and how the system fits within it, user opinions and experiences, which make it perfect for evaluating the early stages of system development [104]. It has similarities with ethnographic studies in that the evaluator must immerse herself in the culture of the organisation. The interviewing takes place as a continuous, free-flowing dialogue with the users. It is based on principles underlying the importance of the context, of the user as a partner, and of a focus for the evaluation study.
- **Interviews:** Involve a discussion between an evaluator and a user, which is useful for high-level evaluation for eliciting user preferences, impressions, opinions and comments, and reveal problems which might have been missed by the evaluator [78]. Unstructured interviews are best used in the early stages of development of the system to gather as much information as possible from the users [170, 222]. Structured interviews, which use a well-defined list of questions, are best used for later stages when answers regarding specific aspects of the system are sought.
- **Focus groups:** Involve a discussion about the system between 6-9 users, and a moderator (the evaluator) [170, 222]. The moderator starts the discussion by using a list of questions, and must guide and facilitate it. This technique can be used both in the early stages of development, to gather requirements, and later, to elicit opinions and comments about the system.
- **Questionnaires:** Users reply to a written list of questions regarding the system [104]. They can be used in any stage of the development process, but are most frequently used post-release.

2.5.1.3 Drawbacks of Involving Users in Usability Evaluation

Despite the recognised importance of involving users in the usability evaluation of software systems, there are also some important drawbacks, both in general, and for

the design of medical software in particular. In describing user testing which, as was highlighted in the previous section, may also involve the use of inquiry techniques, Kieras identified both practical and theoretical limitations [128].

The **practical limitations** are given by the fact that evaluation with users can require time and financial resources which make it infeasible within current software development schedules [128]. In particular, each new iteration of the design requires evaluation by appropriate users using appropriate tasks, and there is never a guarantee that more iterations will not be needed. This issue becomes even more critical for software which is intended for expert users, such as the telehealthcare systems described in this thesis. It has motivated over the years the development within the field of HCI of methods and techniques to speed up iterative design. Examples include paper mock-ups which can be built cheaply for involving users very early in the design, or tools for rapid prototyping, but also the inspection evaluation methods which will be described below.

From a psychological perspective, a **theoretical limitation** is that user testing is not based on human psychology, and does not have a way of representing and documenting the knowledge acquired about the psychology and needs of the users throughout the design process [128]. Such knowledge would be useful for other designs or even for the current design in its current deployment, in case something changes for the user, task, or changes to the design itself are considered. Instead, it is up to the designer's intuition to 'sense' what would work and what would not for the users.

2.5.2 Analytical Methods

2.5.2.1 Usability Inspection Methods

Usability inspection involves a usability expert looking at the design and predicting what effect it will have on typical users, focusing on areas which would cause difficulties to users because of the violation of cognitive principles, guidelines, standards, rules, heuristics or known empirical results [78]. Usability inspections can be carried out from the earliest phase of a design specification. They are cheap, as they do not involve users, and they can help discover relevant problems.

The following are the most important usability inspection methods:

- **Heuristic evaluation:** was described in 1990 by Nielsen and Molich [174]. In 1989 Nielsen had proposed the method as being a “discount usability evaluation” technique ([168]) because at the time the gold standard was to carry out elaborate

and expensive quantitative studies involving many users and complex design representations [172]. Instead, heuristic evaluation allows for early evaluation by a group of experts, who compare a design against established usability guidelines, such as those proposed by Nielsen ([171]), Schneiderman ([201]) or Norman ([177]) [78]. The usability guidelines can also be supplemented with guidelines which are specific to the system domain. Nielsen showed ([169]) that between 3 and 5 experts are sufficient for the method, as 5 experts can find around 75% of all usability problems [78]. They need to work independently in identifying system violations of the guidelines and establishing severity ratings for them according to their frequency, ease of being overcome, persistence and ease of being perceived. The found ratings can then be combined into a severity rating ranging from 0 to 4. Once the evaluators finish their individual evaluation, the mean severity rating is calculated to prioritise the order of correcting usability problems by the design team.

- **Cognitive walkthrough:** was proposed and refined by Polson et al. ([144, 185, 215]) with the purpose of adding psychological considerations to the informal and subjective walkthrough technique [78]. It is based on a cognitive model of learning and use ([215, 103]) and involves experts ‘stepping through’ a sequence of actions needed to accomplish a task on the system for identifying usability problems [78]. The focus usually is on the system’s learnability through exploration, as it was found that users normally prefer to learn a system by exploring it hands on, rather than by being trained or reading documentation. During its preparatory phase, the method requires a fairly detailed specification or prototype of the system, a description of a representative task to be performed on the system, a description of the actions accomplishing the task and a description of the users and their profile [186, 78, 103]. Evaluators use this information to step through the action sequence and provide a story of why the step is or is not appropriate for the user and, where problems are identified, what their severity is [186, 78]. The cognitive walkthrough needs to be documented in detail to make the problems clear to developers and prioritise their correction [78]. A more recent version of the method ([215]) describes each of its steps, to make the method more accessible and usable by evaluators without experience in cognitive psychology. This version requires detailed descriptions of user actions and interface states.
- **Pluralistic walkthrough:** is a version of cognitive walkthrough which was de-

scribed by Bias ([44, 45]) as being used by IBM [186]. It is 'pluralistic' as it involves user representatives, developer representatives and human factors specialists [186, 103]. All the participants assume the role of users within the method and they write down what action they would carry out on each screen which they are shown [103]. When this phase is finished, the participants discuss about each screen, the representative users having priority in the discussion.

- **Formal usability inspection:** was developed by Kahn and Prail ([123]) and involves the evaluation of the users' performance in carrying out tasks on the system by the designers [103]. Designers use their experience, but also models, heuristics and defect detection. The method requires a preparatory phase similar to that of the cognitive walkthrough, and uses a cognitive model of task performance to which one can add the consideration of cognitive steps like the ones in Norman's Action Theory ([176]) [103].
- **Consistency inspection:** involves designers of other interfaces assessing the consistency of the interface with their designs [218].
- **Standards inspection:** involves experts checking the interface's compliance with standards [218].

Despite offering considerable advantages in terms of cost and the possibility of being performed very early in the design of a system, usability inspection methods have some important drawbacks as compared to usability testing. First of all, they do not assess the actual use of the system [78]. Secondly, none of them identifies all of the problems which can be identified by usability testing. Also, they are highly reliant on the expertise of the evaluators. Several studies have found that the usability problems identified by using inspection methods vary considerably between evaluators, which may result in different, unpredictable, recommendations [99, 113, 159, 160].

Heuristic evaluation, the most frequently used inspection technique, requires knowledgeable evaluators, who must be experienced in applying different sets of heterogeneous heuristics and guidelines to different contexts [114, 169, 49, 132, 198]. The heterogeneity of the heuristics and guidelines, and their lack of applicability, even potential contradiction if being applied to a different context, make heuristic evaluation difficult to learn and use. The heuristics and guidelines are often very general and interpretable which may result in different outcomes of the evaluation from different evaluators [114]. Also, evaluators are known for their strong views, so they may focus

on some problems while ignoring others. Heuristic evaluation requires several expert evaluators, who can be difficult to find, and for performing several usability evaluations during the year, which can become costly [103, 115]. As compared to the think aloud protocol of usability testing, heuristic evaluation finds a high number of one-time cosmetic problems, which may not present great problems to the users [114, 49]. The cognitive walkthrough method, which is also frequently used, was found to detect only about one third of the problems detected by heuristic evaluation, but proportionally more severe ones [114, 116]. Although addressing the critiques of its earlier version, the structured content and required detailed description of action sequences of its latest version may limit exploration and hinder the discovery of new solutions to the design [114]. Moreover, the approach usually only considers the user's knowledge of the interface, and not also other user characteristics which could help identify more usability problems [114, 221].

To make usability evaluation more economical, and still identify a reasonable set of problems, usability experts recommend using several different evaluation methods [78, 170].

2.5.2.2 Model-Based Evaluation Methods

Model-based evaluation was proposed as a method for interface design in the seminal work on the psychology of human-computer interaction by Card, Moran and Newell ([62]).

Model-based evaluation methods involve the building of analytical models of the interaction of a user with the system, and their use for obtaining predictions about the system's usability [128]. Kieras described the following important advantages of these methods [128]:

- They allow for the elicitation of results on usability, and the exploration of options for design, even before the first prototype of the system is built and users are involved in its evaluation. This is intended to reduce the need for many iterations and frequent evaluations with users, and thus reduce costs.
- As changing the design to reflect a new system iteration only requires quick changes to the models, they help to speed up any new iterations, unlike evaluation with users (which offers no guarantee in this respect).
- Their results document the reasons behind the level of usability of the design for

their users, which can help in the development of a body of knowledge on the subject.

- They can be easily reused not only for the current design, but also in other designs.

Kieras points out that, unlike other usability evaluation methods, model-based evaluation methods are true engineering approaches which, if used, offer the prospect of quicker, better informed and cheaper development processes, with more usable final products [128].

In a survey of automatic usability evaluation methods, Ivory and Hearst classified model-based evaluation methods such as the ones that I will describe as *automatic analysis* methods, because they can identify usability problems through the use of a computer program [110]. Automation is largely underexplored for other evaluation methods, but it can provide some very important advantages. The ones which were not already mentioned above by Kieras ([128]) are the following, as described by Ivory and Hearst [110]:

- Reducing costs by reducing the time which is spent on evaluation
- Helping to detect usability problems more consistently, based on deviations or usage patterns
- Helping to achieve fuller coverage of the evaluated features of design, as opposed to non-automated automations which limit the coverage due to resource constraints
- Not relying on evaluator expertise, as do inspection methods
- Allowing for thorough comparisons between alternative designs

In the following subsections I will describe some more recent, and well known, model-based evaluation methods which can be used to predict user behaviour especially in terms of task completion time, and which are thus relevant for this thesis.

2.5.2.2.1 GOMS

Card, Moran and Newell ([62]) proposed in 1983 the *Model Human Processor (MHP)*, a psychological model of humans consisting of a perceptual, motor and cognitive interacting systems, each with its own memory and processor [186, 73]. They also

presented **GOMS (Goals, Operators, Methods and Selection Rules)** as an engineering model to be used for applying the characterisation of human performance made by MHP to task analysis.

GOMS was intended for providing “engineering models of human performance” ([121]), which can produce with little effort quantitative performance predictions with different levels of approximation, prior to prototyping and user testing [121, 73]. Making relevant predictions on systems which have not been totally built requires, as John and Kieras suggest, “parameters that are robust and reliable across tasks and can be used without further empirical validation”, which is why different authors have proposed tables of parameters for different tasks ([62, 91, 178]) [121]. Apart from predicting human performance, Kieras explains that GOMS can be used qualitatively to evaluate the design’s completeness, naturalness, consistency and efficiency and provide suggestions for improving the design ([127, 186, 120]), as well as to design training, help or documentation material [120]. Moreover, the depth of the goal decomposition can estimate short-term memory requirements [78].

GOMS starts from the notion of a goal, which is what the user wants to achieve [121, 78, 73]. **Goals** are considered as points in the user’s memory to which the user can return should any error occur. They are often decomposed into subgoals [78]. **Operators** are the basic actions (perceptual, motor or cognitive), affecting either the system or only the user’s mental state, which must be performed by the user on the system in order to achieve the goal [121, 78, 73]. They constitute the lowest level of analysis and can be considered at different levels of abstraction, from the command level down to the keystroke level [78]. **Methods** are well-learned options for decomposing a goal into subgoals among which the user can choose [121, 78, 73]. **Selection rules** define rules for choosing between several methods for accomplishing a goal [186, 121, 78, 73]. The notions of goals, operators, methods and selection rules are used to define the procedural knowledge required from a user for performing a task [186, 121]. Different versions of GOMS, with different capabilities, but based on the same notions, have been developed over the years:

1. **CMN-GOMS (Card, Moran and Newell GOMS [62])**: the original version of the approach. It explains how goals, methods and operators can be represented and how selection rules can be formulated, where goals need to be placed hierarchically, operators are executed sequentially and methods are represented in an informal pseudo-code which may contain submethods and conditionals [121, 120].

2. **KLM (Keystroke-Level Model [61])**: uses only the lowest level subgoals (keystroke-level operators) for making predictions, for which time estimates can be calculated [121, 120].
3. **NGOMSL (Natural GOMS Language [126])**: provides a more natural, programming language-like notation of the GOMS components, as well as guidelines regarding the number of steps in a method, the setting or ending of goals and user memory requirements during task accomplishment [121, 73].
4. **CPM-GOMS (Cognitive Perceptual Motor, or Critical Path Method GOMS [119])**: allows the representation of tasks within which operators are performed in parallel by means of cognitive, perceptual and motor operators within a PERT chart [121, 120]. It is based directly on MHP, thus on the model of parallel multiprocessor or human information processing [120].

While all of the GOMS methods can decide on the effectiveness of a system and predict user performance, CMN-GOMS and NGOMSL also help predict the order of operators which users will choose for a task [121]. Moreover, NGOMSL can also predict learning time, the advantages of consistency, and the chance for some errors.

Due to some initial criticism about the difficulty and tediousness of doing task analysis and time calculations by hand ([41, 58, 106, 129]), tools have been developed for automatically creating GOMS models (e.g. Log Analyzer [180]), and obtaining quantitative predictions from them (e.g. USAGE [58], CRITIQUE [106]), or only for obtaining quantitative predictions from an existing GOMS model (e.g. GLEAN [129]) [110].

2.5.2.2.2 PUM

PUM (Programmable User Models) have first been described by Young et al. ([220]) as user models which, by being given basic human problem solving skills and initial knowledge, would help understand how a human uses her knowledge when interacting with a system (here named device). This could explain, for example, what knowledge users need for the interaction, whether users have this knowledge or the system needs to be redesigned, how this knowledge was acquired by the users, whether it is rational to expect users to have this knowledge, etc. The modeller must extend the user model so that it interacts with a model of the device.

Blandford et al. have worked on extending the original PUM approach ([54, 46, 47]). In a more recent paper ([47]), they proposed an extension which considers that humans act rationally in accomplishing their goals, according to the *principle of rationality* of Allen Newell: “If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action” [163]. The approach uses a framework of human cognition and problem-solving, which was derived by the authors from the literature and empirical results [47]. The framework sets out rules on how humans use and apply their knowledge in the achievement of their goals. In particular, the user’s knowledge is made out of **beliefs**- what she knows about the world, and **operations**- how she can “change the state of the world” [47]. Each operation has an associated device action, and defines the user’s beliefs about the circumstances under which the device action can be committed to, and why, with what effects, and under what circumstances it can be performed on the device. The user also has a **goal**- the belief to be acquired. The device is defined by means of a **device state**, a set of permitted **initial states**, and a set of permitted **actions** which can change the device state. The user behaves in a goal-directed way, by picking an operation which would at least partially achieve her goal and to which she can commit. If the operation also meets the criteria for being performable, its action is invoked on the device. This will lead to the device changing its state, and the user updating her beliefs according to this new device state and checking consistency with the operation’s expected effects. If instead the operation does not respect the criteria for being performable on the device, the user commits to it (i.e. makes it a **commitment**), and sets herself the subgoal of achieving that criteria. If an operation which she has committed to (i.e. commitment) becomes performable, the user checks if she does not believe in its effects already and, if not, performs it on the device and drops the commitment. If she believes in its effects already, she can drop the commitment directly.

Operational models instantiate the framework with an empirically-derived scenario of a real system’s use [47]. The results of running the operational models indicate user knowledge needs and whether the user’s goals are achievable or not, a negative result suggesting a usability problem. The analysis of the operational models can also help the modeller detect sources of human error and decide on the efficiency of the interaction. The approach models experienced users, with the purpose of identifying usability problems which would become even more critical for beginners. Abstract models can be built from operational models to test stronger properties of the identified user behaviour by means of formal proofs.

2.5.2.2.3 Cognitive Architectures

The different types of GOMS methods have been described by their authors as being based on different simplified cognitive architectures, as they allow the identification of some design issues without the need for the modeller to delve into theoretical considerations [128]. A PUM (Programmable User Model), on which the work of Blandford et al. ([54, 46, 47]) is based, is also defined by Young as a “psychologically constrained cognitive architecture” [220]. Cognitive architectures provide inspiration for a more detailed and better validated representation of the cognitive constructs within a brain, thus allowing us to build user models which are closer to the reality of human cognition.

A cognitive architecture can be defined as “a broad theory of human cognition based on a wide selection of human experimental data, and implemented as a running computer simulation program” [57]. The notion of cognitive architectures appeared ever since the beginnings of artificial intelligence (AI) and cognitive psychology (e.g. the General Problem Solver approach presented by Newell and Simon [165]) and was motivated by Newell in the book “Unified Theories of Cognition” ([164]) [57].

A simple Google search on cognitive architectures revealed over 20 names, with only very few of them being explicitly used for usability, for predicting problems related to performance in terms of error/learning rates and the transfer of knowledge. Cognitive architectures are used more for modelling agents for simulated training environments, for building computer tutoring systems or interactive computer games. I will describe the three most relevant cognitive architectures for the purposes of this thesis: Soar ([134, 143, 133]), ACT-R ([34, 35]) and ICARUS ([139, 67, 137, 140, 142]).

2.5.2.2.3.1 Soar

Soar ([134, 143, 133]) has been under continuous development, having had nine major versions since 1982. Some of its important uses were:

- Exploring user learning needs ([32, 31]) and building models of speeding up exploratory learning ([189, 105]), which led to the derivation of design guidelines: reducing display clutter and structuring information ([31]), using good labels ([189, 105]), detecting pauses of gaze and offering suggestions on what menus lead to ([189]) [57]

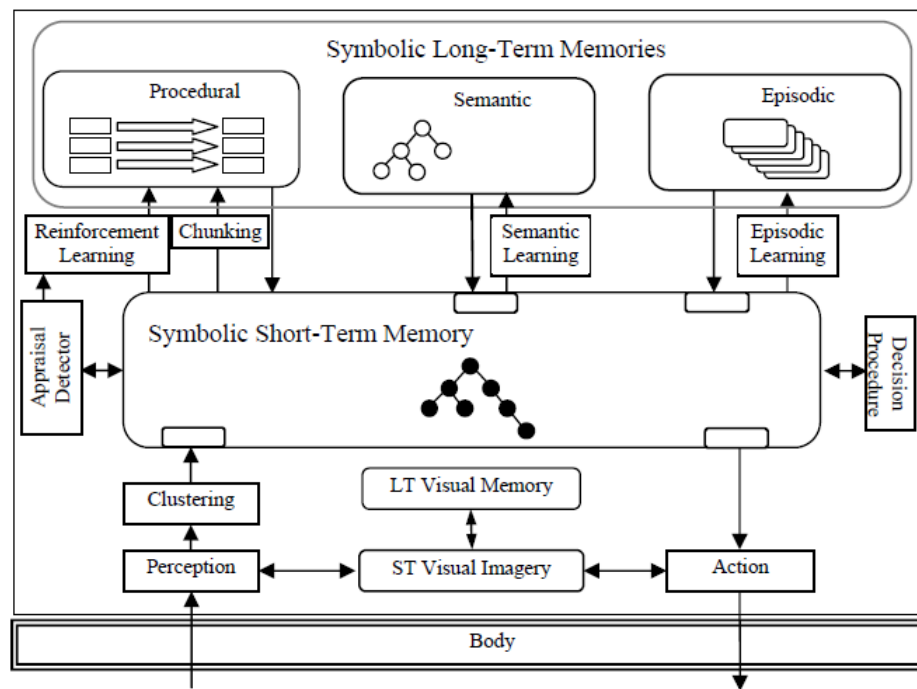


Figure 2.1: The Structure of Soar 9, taken from [133]

- Modelling elements of human language processing ([145]) or categorisation ([156, 141])
- Constructing simulated agents for the testing and preparation of a Space Shuttle ([162, 57])
- Constructing simulated agents in wargames for military training (Tac-Air [122]) [57]
- Constructing underlying intelligent agents for characters in computer games ([148, 141])

A representation of the basic structure of Soar 9 ([133]) is provided in Fig. 2.1. Soar is purely symbolic and essentially uses two types of memory, as described by its authors [133, 143]:

- **Long-term memory (LTM):** knowledge that exists independent of the current situation [133, 143]. It is represented as production rules and consists of:
 - **Procedural knowledge:** how and when to do things. It controls behaviour by mapping to operators [143].

- **Semantic knowledge:** what is known to be true. It is used for solving impasses when procedural knowledge is incomplete or not useful for attaining the current goal [143].
- **Episodic knowledge:** things which are remembered. It is also used for solving impasses [143].
- **Working memory (WM):** represented as a symbolic graph structure containing pairs of objects and their properties and relations which make up the current state [133, 143].

To populate the WM, Soar uses **perception** from the environment or retrieval from LTM [133, 143]. An action in the environment is performed by initiating **motor commands** in a buffer in WM [133].

Added to this basic structure which is traditional for Soar, Soar version 9 added new non-symbolic knowledge representations, learning and memory modules ([133]), but I will not describe them here as they are not important for this thesis.

The functionality of Soar is based on characteristics of cognitive behaviour defined by Newell in his book, “Unified Theories of Cognition”([164]) [143]. It is basically a production system, but it can also be seen at a more abstract level, being based on Principle P9 of the Model Human Processor ([62]), the **problem space principle**, and the general idea of movement through problem spaces ([175, 166]): goal-directed behaviour is characterised by the use of operators to move through problem spaces for reaching a desired state [143, 57].

Soar’s operation is cyclic, with the basic cycle named a **decision cycle** [143]. In each such cycle, Soar generates behaviour out of memory contents, with the aim of selecting the next operator to apply. The steps of a decision cycle are the following [143, 133]:

1. **Input:** WM elements are created that reflect changes in perception;
2. **Elaboration:** the contents of WM are matched against the “if” parts of the rules in LTM. All rules that match from procedural memory fire in parallel, leading to changes to the current goal and state, and generating suggestions or preferences (symbolic or numeric) for operators in WM.
3. **Decision:** the preferences are compared and an operator is chosen. If Soar cannot choose between operators, an impasse is generated and Soar automatically creates a substate with the goal of solving the impasse (**problem solving**).

4. **Application:** the operator's actions are carried out in parallel or sequentially by matching rules.
5. **Output:** any issued commands are forwarded to the motor system.

Learning is an important feature of Soar. Soar uses several learning mechanisms:

- **Chunking:** the most developed learning mechanism, which converts the results of problem solving into new rules in LTM such that in the future a similar situation does not cause a new impasse [143]. The superstate used for the impasse becomes the condition of the new rule and the solution to the impasse becomes the operator decided on.
- **Reinforcement learning:** adjusts the numeric preferences of operator selection rules in an attempt to maximise reward [143].
- **By using episodic memory:** snapshots of WM elements are saved automatically in episodic memory when an action is taken in the world [143, 133]. Impasses can be solved by creating cues to retrieve such snapshots (episodes), and thus learn from similar past experiences.
- **By using semantic memory:** declarative facts about the world are saved in semantic memory [143, 133]. Data chunking can be used for learning appropriate rules through retrieval from semantic memory, by forcing an impasse when known structures in WM occur.

Retrieval from episodic memory is influenced by activation, as WM elements are given activation based on when (recency) they matched rules that fired and the relevance of these rules [143].

2.5.2.2.3.2 ACT-R

ACT-R (Adaptive Control of Thought-Rational [34, 35]) has been under continuous development since the late 1970s [141]. It was used primarily to model phenomena from experimental psychology, to create models in domains such as learning and memory, attention, perception, reasoning, problem solving, language processing, cognitive development or individual differences [141, 92]. Several studies have demonstrated fits between the behaviour of the architecture and quantitative data on human error rates

and reaction times. In HCI, ACT-R was used for producing user models for evaluating different computer interfaces (e.g. SnifAct, which simulates a user's navigation behaviour according to information scent [182, 92]). Other uses of ACT-R included:

- Building cognitive tutoring systems which deduce difficulties students have and provide help, with wide use in schools (e.g. [131]) [141, 92]
- Building cognitive agents for training environments (e.g. [42]) [92]
- Interpreting fMRI (Functional magnetic resonance imaging) data for neuropsychology (e.g. [33])[92]
- Building mobile robots that interact with humans (e.g. [208]) [141]

ACT-R 5.0, the architecture of which is represented in Fig. 2.2, consists of a series of modules, each with its own purpose: a **visual module** (for visual processing), a **manual module** (for controlling the manual performance of actions in the environment), a **declarative module** (for storing and retrieving long-term known facts), and an **intentional module** (for monitoring current goals and intentions) [35, 141]. A **central production system** stores long-term procedural knowledge (i.e. how to do things), in the form of production rules, and coordinates the modules. It is not influenced by the modules' activity, but only by the small quantity of information stored in their buffers. **Buffers** are the modules' only interface to the central production system, the information contained at one time in all the buffers forming the current state (short-term memory STM) [35, 141, 92]. Each buffer can only store a single relational declarative structure named a 'chunk', which is different from Soar's chunks. The different components of the architecture are mapped to real areas of the brain, as explained in [35].

On each processing cycle, production rules from the central production system which match chunks from buffers fire [35, 141]. This changes buffer contents, leading to actions in the modules such as executing a motor action in the environment or retrieving information from LTM. The processing is a mix between parallel in modules and serial in buffers (only one chunk of information can be stored at a time) and the central production system (only one production rule can be selected to fire at one time) [35].

The ACT-R cognitive architecture is considered **hybrid**, as it combines a **symbolic structure**, represented by the production system, with a **sub-symbolic structure**, represented by a set of parallel processes controlling many of the symbolic processes [92].

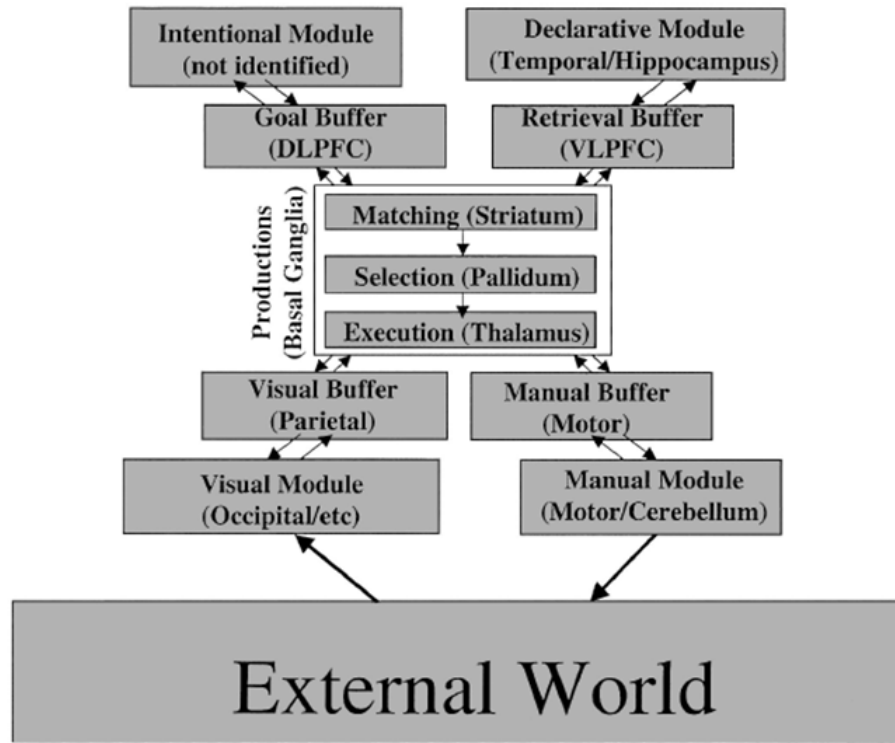


Figure 2.2: The Structure of ACT-R 5.0, taken from [35]

For declarative memory, a sub-symbolic equation calculates the **activation of a chunk** as being the sum of the **base activation (how useful it was in the past)** and **associative activation (how relevant it is now)** [35, 92]. The activation influences if and how fast the chunk will be retrieved from declarative memory. For procedural memory, another sub-symbolic equation calculates the **utility of a production rule** based on the probability for it to achieve the current goal (deduced from past experience), the value and relative cost of achieving it. The production with the highest utility (if noise is also considered) is selected at once.

Learning in ACT-R is also based on sub-symbolic mechanisms, being both structural and statistical and closely related to the equations described above: the use of declarative chunks by production rules increases their base activation, which otherwise decreases (simulating forgetfulness), while the utility of production rules is updated based on their observed behaviour [35, 141].

ACT-R has traditionally focused on cognitive capabilities, while being weak in perceptual and motor ones [10]. ACT-R-PM ([56]) was developed as an extension in this respect [10]. It separates out a cognition layer, consisting of the procedural and declarative memory and behaviour of ACT-R, from a perceptual/motor layer, which

stands between cognition and the environment and consists of well researched perceptual and motor modules. ACT-R/PM has been used for predicting search times in graphical user interfaces [86, 85]. Ritter et al. have proposed building embodied models in ACT-R/PM which can directly interact with an interface to predict its efficiency, user knowledge requirements, and make it possible to compare interface designs [190, 191]. Li and Gunal have more recently promoted the use of cognitive modelling through ACT-R/PM, and its extension to training/calibrating user models by using empirical human subject data and automatically running them on the interface, for deciding between alternative designs during the design stages of in-car systems [146]. Unfortunately, I only identified the previous papers at a stage when it was too late to use them as inspiration for my work.

2.5.2.2.3.3 ICARUS

ICARUS ([139, 67, 137, 140, 142]) is a more recent cognitive architecture. Although it was intended as an architecture for physical agents, it was built following, and extending, classical theories of human problem solving [138]. ICARUS was used for:

- Developing agents for well-known tasks from AI and cognitive science (e.g. [139, 141])
- Developing synthetic characters for simulating in-city driving ([67, 137]) and first person shooter games ([68, 141])
- Building physical robots that work together with humans (ongoing work) ([141])

Like other cognitive architectures, ICARUS distinguishes STM constructs, which are more dynamic, and LTM constructs, which are more stable [142]. As opposed to other cognitive architectures and only informally stated by ACT-R ([34, 35]), ICARUS explicitly states that elements in STM are instances of general elements in LTM.

Each of the memory constructs is further split into several types of memory, as described by Langley et al. [142]:

1. STM, consisting of:

- A **perceptual buffer** storing **percepts**, representing the objects from the environment that the agent perceives on one cycle. In a telehealthcare scenario, an agent might perceive, for example, a patient and her readings, each with their attributes.

- A **belief memory** storing **beliefs**, representing “higher level inferences about the agent’s situation”, which are to be deduced by the agent as opposed to being provided as input by the modeller. Beliefs normally represent inferred relations between percepts, but may sometimes also refer to an inferred property of a single percept. For example, in a telehealthcare scenario an agent might deduce as beliefs that two perceived patients suffer from the same condition, or that a perceived patient needs to be monitored. Each belief from belief memory must be an instance of the more general definition of a *concept* from *conceptual memory* (the notions of conceptual memory and concepts will be described below). Additionally though, the modeller may add to belief memory beliefs which will never change because it is more efficient to have them as static instead of having the agent infer them repeatedly.
- An **intention memory** stores **intentions** describing the activities that the agent wants to carry out as being possible for him in the environment. For example, the agent might decide to monitor a patient, and thus this will be translated into an intention. Each intention must be an instance of a more general *skill* from *skill memory*, which will be defined below.

2. LTM, consisting of:

- A **conceptual memory** storing **concepts**, the definitions of beliefs. Concepts describe the general inferences the agent can make about the relations between certain percepts or the properties of certain percepts. For example, in a telehealthcare scenario the agent could have a concept which dictates when two perceived patients (percepts) would suffer from the same condition. When the agent’s perceptual buffer includes (the agent perceives) two patients respecting these conditions, the concept becomes applicable and it is instantiated into a belief about these two patients suffering from the same condition. In the same way, a concept dictating when a patient is alerting becomes applicable when the agent perceives a patient respecting these conditions, and will be instantiated in a belief about that certain patient alerting. Concepts can be non-primitive, if they require that other concepts are also applicable or not before they become applicable themselves, or primitive if they don’t depend on other concepts and only directly check the existence of certain percepts, their relationships or properties. They thus

form a hierarchy, with more and more complex, non-primitive, concepts placed at the top, and primitive concepts as the last elements at the bottom.

- A **skill memory** storing the **skills** that the agent knows he has available for achieving his goals. Skills represent the activities that the agent can pursue. They define conditions in terms of beliefs that the agent should or should not have in his belief memory regarding certain percepts such that either an action in the environment, or certain subskills, should be pursued. Skills have certain expected effects in terms of beliefs that should be present or not in belief memory if the skill is accomplished, and a skill is not applicable if its effects are already met. For example, in a telehealthcare scenario the agent could have a ‘monitor_patient’ skill which dictates that a perceived patient (percept) should be alerting, in order for the agent to call subskills necessary for monitoring the patient. These subskills could for example refer to selecting the patient, checking her readings, and entering conclusions on the system. The (top) skill’s expected effect could be that the patient will no longer be alerting in the end. This skill becomes applicable when the agent does perceive a patient who is alerting, and as a result it is instantiated into an intention. Like concepts, skills can be non-primitive or primitive. Non-primitive skills require the achievement of subskills, while primitive skills do not depend on other skills, but require the direct achievement of an action in the environment, in order to be themselves achieved. In this way, skills form a hierarchy. As skills are instantiated into intentions, intentions will also form a hierarchy. The architecture can only process one intention at once, and thus each intention will keep a reference to its parent intention.

Skill execution is considered in ICARUS as the basic building block which allows the agent to solve complex problems needing the composition of skills [142]. The following are the additional notions that are necessary for problem solving:

- **Goals** describe “some aspect of a situation that the agent considers desirable” [142]. They define those beliefs regarding certain, or any, percepts that should/should not end up for the agent in his belief memory, for satisfying them. For example, the agent’s goal for the above examples could be to end up with a belief about a certain patient not alerting (a negation about the patient alerting), or having been monitored (if a belief instantiating such a concept can be derived).

- **Problems** keep track of the agent's progress towards achieving her goals, in terms of achieved goals and goals which are yet to be addressed, and the agent's next intention for addressing the unachieved goals [142].

ICARUS performs the following types of processes [142]:

- **Conceptual Inference:** On each cycle, the agent perceives elements from the environment and adds them as percepts to his perceptual buffer. The existence of these percepts, relations between them and these percepts' properties may lead to the applicability of certain concepts, which thus get instantiated into beliefs. As concepts may depend on one another in their hierarchy, this may lead to the applicability of other concepts from higher in the hierarchy, which also get instantiated into beliefs. This process continues until no more concepts are applicable.
- **Skill execution:** Given a skill that the agent wants to pursue, the architecture will:
 1. Perform conceptual inference to generate beliefs out of concepts which become applicable given current percepts.
 2. Find skill(s) which match the wanted skill and for which there are applicable percepts.
 3. Decide which skill(s) (if any) are applicable (their conditions are respected and their effects are not already met) and select one skill at random.
 4. Create an intention by instantiating the skill.
 - (a) If the intention has subskills (it is non-primitive), attempt to perform the subskills (go back to 2).
 - (b) If not (it is primitive), carry out the intention's action in the environment.
 5. If the beliefs obtained from a new conceptual inference confirm the intention's wanted effects, mark the intention as successfully completed.
- **Failure and recovery:** The agent detects failure within the skill execution process if it does not find an applicable skill for an intention's first/ n-th subskill, or it goes on for a certain number of cycles attempting to achieve a subskill without success. For these cases, the intention is marked as a failure for the parent intention such

that the parent intention looks for alternatives without considering that intention again. If the top intention is determined to be a failure, the architecture abandons the task entirely.

Problem solving is another important process in ICARUS [142]. It involves the agent identifying those goals within a set of goals which have not been satisfied yet, and attempting to address them by heuristically choosing between skills, instantiating them into intentions and performing the intentions. Skills address goals if the beliefs from their expected effects match with the beliefs of the goals. Intentions are performed as in the points 4-5 of skill execution from above, if their conditions are respected. If not, the agent creates new goals to attempt to bring about the beliefs from the conditions. If an initial goal is unachievable by any skill and refers to beliefs which would be instantiated from non-primitive concepts, the agent creates new goals to attempt to bring about the subconcepts. Problem solving therefore allows the agent to address her goals even if they are not directly achievable by her skills. Problems are used throughout the process to keep track of the goals which have or have not been achieved, and any percepts from the perceptual buffer which were used. The process finishes when the top problem is solved. Backtracking and recovery from failure are used throughout to help the agent identify inapplicable intentions, goals at a certain step or subproblems such that they are not tried again for the same problem.

2.5.2.2.4 Choosing a Model-Based Evaluation Method

The model-based evaluation methods presented in this section all have their advantages and disadvantages.

GOMS ([62]) is a well researched and validated family of methods, with a track record for success [128]. The different GOMS methods offer the great advantage that they can be used easily, at a high level or informally, early in the detailed design of a computer system, when quick design decisions are required. Nevertheless, they also have their disadvantages, the greatest one being that they are meant for modelling the routine, optimal behaviour of experienced users, and cannot deal very well with creative work and unpredictability.

Cognitive architectures, on the other hand, can be used for modelling uncertainty, and a human's problem solving and learning processes for tackling with it [128]. They have strong foundations in psychological theories and empirical results, and support the modelling of human cognitive functions in detail. This offers greater

flexibility for predicting different usability characteristics, including efficiency and erroneous behaviour. Their greatest disadvantages are their theoretical complexity, and the difficulty of building the models.

Most of the work presented in this thesis could have been achieved by using a GOMS ([62]) method, but I chose instead to get inspiration from cognitive architectures due to their greater flexibility. The main purpose of this thesis is to use such approaches for post-development, pre or post-deployment, evaluation, and not during design stages. In such a case, it may be worth investing in a more complex model-based evaluation method such as cognitive architectures, for being able to obtain more varied and accurate usability characteristic predictions, but also a more detailed understanding of user behaviour. Moreover, my vision is that, with experience, model inputs could be reused for different deployment contexts, and thus speed up model development.

I first implemented a modelling approach inspired by PUM, the framework of which is a “psychologically constrained cognitive architecture” ([220]), but my results had some important limitations, as will be described in Chapter 5. I therefore decided to use as inspiration the ICARUS ([139, 67, 137, 140, 142]) cognitive architecture, as I will describe in Chapter 6.

My intention was to develop a modelling approach which can be used to represent complex cognitive processes, as the ones which are involved for making treatment decisions by doctors. Although my focus was on efficiency, the main functionality of the approach can also be used to predict other usability characteristics, as I will explain in Chapter 10.

2.6 Practice of Usability Evaluation for Medical Designs

Despite the advantages of considering usability in the design, developers usually focus too much on functionality and too little on usability in telehealthcare solutions [202]. In fact, usability has only been considered at a very late stage in the history of medical informatics [206]. End-users are often not involved enough in the design of medical systems due to a commercial urge for a speedy development, which leads to the discovery of usability problems only after roll-out, when they are more expensive to address and could lead to the rejection of the system by its users [179].

As I observed in my previous work ([25]), it is often the case that telehealthcare solutions are used as off-the-shelf, without being adapted to their intended users in each new deployment first. This somehow contradicts the traditional nature of medical work,

which lacks golden standards as to how patients should be diagnosed and treated and varies greatly from country to country, practice to practice and even among individuals. This results in usability problems which cause frustration, inefficiency, and can put patient safety at risk.

While it is generally true that usability problems cannot be foreseen in the absence of a proper pre-deployment usability evaluation, this issue becomes greater if the system is to be used in various contexts which differ in key elements such as user characteristics, workload or patterns of use. During two post-deployment usability evaluations of telehealthcare systems that I performed as part of my Masters ([25, 27]), I found that usability problems may manifest differently across, and even within, healthcare organisations. In particular, user perception about usability was influenced by the number of patients and their characteristics, user roles and the frequency of their work on the system, and even the speed of the available internet connection.

2.7 Other Related Work

2.7.1 From Systems Engineering- Modelling and Simulation

The approach that I will describe in this thesis uses modelling and simulation techniques, and therefore it is important to briefly describe modelling and simulation.

Modelling and simulation is often used in Systems Engineering for deciding on requirements, analysing or testing a system or process [6].

Modelling is the process of creating a model- a simplified version of some system of interest- which can be used to understand the system [36]. A model incorporates the system's features to investigate, and should be easy to experiment with. **Physical models** are simplified representations of a system's physical characteristics [6, 12]. **Mathematical models** use mathematical terms and equations to translate the behaviour of a system, and the relationship between its components. A **simulation model** can be mathematical or physical. It is operated by a **simulation** for the purpose of predicting certain measures of interest for the real, planned or existing system. Simulation models are usually *stochastic* (either their inputs, outputs or both are probabilistic) and *dynamic* (interaction over time is considered) [36]. The process of building a simulation model of a system is provided in Fig. 2.3 [12]. Here, the conceptual model is the model which incorporates the factors which we hypothesise would affect the measures of interest.

Simulations with changed model configurations can be performed to explore the

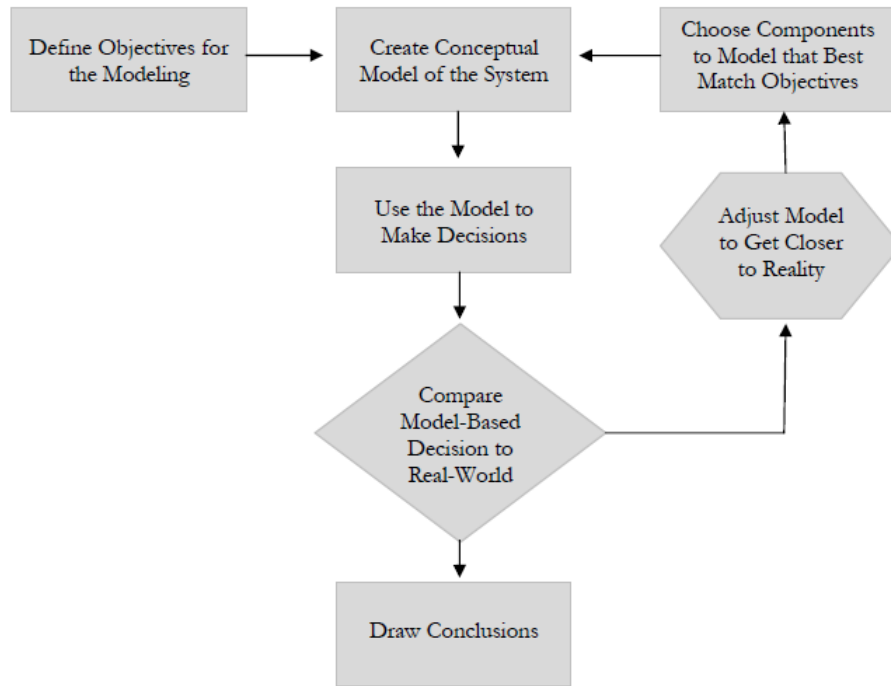


Figure 2.3: Process of building a simulation model, taken from [12]

effects on measures of interest of ‘what-if’ changes to the real, planned or existing, system [36]. To do this, once a simulation model is built, we first choose the experimental design, in terms of the measures to assess and the model configurations (set of variables and values) which we hypothesise would influence it. We must then decide on experimental conditions such that accurate and rich information is obtained from each simulation run. Finally, we run the simulations with the different configurations. To analyse the results, we build statistics and graphical displays for the data, test the hypotheses against them, document the results and conclusions, and decide on a further course of action.

Modelling and simulation offers important advantages. As described by Anu, it helps [36]:

- Better understand a system.
- Test hypotheses without incurring the risks and costs of implementing or changing a real system.
- Experiment with new or unknown situations if only partial data is available.
- Plan for and develop more robust systems.

- Reduce development times, and thus also costs.

2.7.2 From Scientific Management

Like my work, the Scientific Management theory is concerned with the analysis of efficiency in a work context [207]. Its purpose is though different- that of improving labour productivity through the systematic improvement of the time taken for different tasks, for maximising economic efficiency. Scientific management was founded at the beginning of the 20th century by Frederick Winslow Taylor in his book “The Principles of Scientific Management” ([207]), as a way of dealing with inefficiencies in industrial processes. Despite not being in use in modern management any longer (it was criticised for leading to an excessive division of labour and the de-motivation of staff), some of the methods and techniques of scientific management are still used today for analysing job designs [203].

Scientific management was divided into two fields of study: *method study* and *work measurement*, which are both generically referred to as *work study* [203].

Method study (also known as methods engineering or work design) was defined by the British Standards Institution as “the systematic recording and critical examination of existing and proposed ways of doing work, as a means of developing and applying easier and more effective methods and reducing costs” [8]. Method study critically examines facts about the currently used methods for a job and determines new methods which would make work more efficient by removing unnecessary steps, delays and the waste of human or material resources [151]. **Motion study**, defined by the followers of Taylor, Frank and Lillian Gilbreth ([90, 188]), and then extended by Barnes ([40]), Maynard ([152]) and others, is a method used as part of this field. Its purpose is that of reducing inefficiencies in work by analysing and reducing motion [147]. It was applied to healthcare and life sciences ever since 1914 ([90]) [147].

Work measurement, also called *time study*, was defined by the British Standards Institution as “the application of techniques designed to establish the time for a qualified worker to carry out a specified job at a defined level or performance” [151]. Work measurement aims to reduce work inefficiencies by analysing and reducing wasted time and establishing time standards for types of work [151]. The most used methods for work measurement are time studies (stop-watch technique), and work sampling. **Time studies** were initially proposed by Taylor ([207])[147]. They involve the observation of workers using a stop-watch to record the times and rates of working on specific

tasks, under certain conditions, and the analysis of the results to decide on the time required for the task for a certain level of performance. **Work sampling** involves the performance of a high number of observations within a fixed time to count the number of occurrences of a particular activity or delay, and decide on the percentage of times in which they take place [151].

Time studies and motion studies have been integrated as Time and Motion Studies, which involve both the observation of what a person is doing and for how long. Time and Motion studies are considered the most reliable scientific management method for analysing cost, inefficiencies and the effects of change in healthcare organisations [147, 224, 84, 53].

2.7.3 From Project Management

Following in the footsteps of Taylor and his students, Henry Gantt and Henri Fayol, project management has developed into a discipline since the 1950s [217]. The most important mathematical project modelling techniques used in project management are the Critical Path Method (CPM, also known as Critical Path Analysis, CPA [125]), and the Program Evaluation and Review Technique (PERT [150]). They represent ways of identifying, analysing and organising the activities and resources which are necessary for a project, such as to ensure that the project is finalised on time and with appropriate costs. Although at a much higher level, they are used, like my work, for analysing and optimising time. I will therefore briefly describe these two techniques.

CPM was developed in 1959 by DuPont Corporation and Remington-Rand ([125]). The process of building a CPM involves the identification of the activities which are required for a project, their sequencing, duration, resource allocation and dependencies, taking into consideration any milestones or deliverables [158]. They are represented graphically, usually within an activity-on-node diagram. By using what is called a *forward/backward pass procedure*, the method is used to determine the optimal earliest start and finish times of each activity. The *critical path*, the route through project activities in which there is no wasted time (called *total float*), is then calculated. The critical path represents the longest time in which all of the required activities of the project will be computed, and the project's shortest time. A delay in any of the activities along the critical path (also called *critical activities*) will lead to a delay of the whole project.

CPM is useful for projects where activity times are deterministic [100]. For those

where they are imprecise, **PERT** ([150]) is a better choice of technique, because it uses probabilistic times [100]. As compared to CPM, PERT can inform better about the probability for the project to be delivered on time. It was first presented by the US Navy in 1958 [150]. The process of using PERT starts with identifying activities, their sequence and milestones [71]. Then, the activities and events are represented in a diagram, either an activity-on-node diagram (like for CPM), or more frequently an activity-on-arrow diagram. The next step is to decide on *optimistic*, *pessimistic* and *most likely* times for each activity, according to a *beta distribution* [71, 150]. For each activity, the expected time and time variance are calculated and then added, together with the rest of the details, to a table of sequentially ordered activities [71]. Like for CPM, the *critical path* of the diagram is identified through forward and backward passes. Finally, the *central limit theorem* is employed to calculate the probability for different project durations.

Chapter 3

Usability Study of a Telehealth System Used in Lothian, Scotland

3.1 Overview

Following lessons learned from a major telehealth project as part of the Telescot Programme ([21]), NHS Lothian started a collaboration with NHS 24 in early 2012, with the purpose of developing a mainstream telehealth service for the home based telemonitoring of patients suffering from COPD (Chronic Obstructive Pulmonary Disease) and CHF (Chronic Heart Failure). The service development was gradual, and its continuous evaluation, including that of the technology, was seen as key both for it becoming mainstream in Lothian, but also for contributing lessons for potential scaling up as part of the DALLAS programme ([23]). During the period February-June 2013, I got involved in the evaluation by aiding with the assessment of the usability of a new telehealth system which NHS 24 had moved to since September 2012. My study consisted of two parts involving the system's different users: the NHS 24 call operators (COs) and their supervisors (first part of the study, February-May 2013), and the following care provider teams (second part of the study, May-June 2013):

- The Edinburgh Community Respiratory Team (CRT): respiratory physiotherapists monitoring patients with COPD during weekdays and weekends
- The East and Midlothian COPD Team (EMACS): nurses monitoring patients with COPD during weekdays only
- The West Lothian Intensive Case Management Team: nurses monitoring patients with COPD during weekdays only

- Lothian Unscheduled Care Service (LUCS): doctors monitoring patients with COPD or CHF during weekends
- Heart Failure Nurses, monitoring patients with CHF during weekdays only

In this chapter, I will present the main results on usability of this study, and the suggestions that users had. As we will see throughout the chapter, the great majority of identified usability problems were either efficiency problems, or would affect efficiency. The aim of this chapter is therefore to emphasise the importance of efficiency for the users of telehealthcare systems, and to motivate my work in developing approaches for automatically predicting it.

The data collected during this study was also used for deriving inputs for the approach for predicting efficiency described in this thesis. I will not refer to this aspect of the study in this chapter (interested readers are referred to Chapter 8), but only to the qualitative findings on usability.

3.2 The Telehealthcare Service

By using an application running on a tablet installed in their home and wireless measurement devices, patients could take physiological measurements (e.g. of oxygen saturation and pulse for COPD) and answer a daily health survey. The measurements were sent by Bluetooth to the tablet and then, together with the patients' survey answers, via the Internet to a remote secure server. Alerts were computed based on pre-existing algorithms which attributed scores to symptoms, or if physiological measures were outwith previously agreed parameters. All of the resulting data was made available to monitors on a telemonitoring website. Every day, one non-clinical NHS 24 CO would monitor patients on the website between 9 am and 1 pm. She would complete an Excel spreadsheet with each patient's alerts and take an appropriate action as directed by it. These actions could be: call the patient to find out more about her health and, if necessary, to ask her to retake her measurements for a reassessment; send an email to the patient's care provider team, attaching the spreadsheet, if the patient's state was not improving or getting worse, or if she had not succeeded in contacting the patient (Contact Care Provider outcome); or conclude that the patient did not require an intervention (No Action outcome). The NHS 24 CO would also record at the end of her shift statistical data (the number of patients monitored that day, the outcome of her monitoring, frequency and duration of calls, etc.) by using a separate table, also an

Excel spreadsheet. If alerted by the CO email, the patient's care provider would log-on to the telemonitoring website to reassess the patient's condition, and contact the patient to discuss her health, offer her advice and change her treatment if necessary. In addition, the care provider would sometimes need to use the patient's paper or electronic record to record data about the patient, prescribe medication or obtain patient test results.

During the deployment, NHS Lothian and NHS 24 suggested some changes that would make the telemonitoring website and the Excel spreadsheet used by monitors more useful for monitoring the two conditions and easier to use. The company producing the software provided a new version of these applications, Version 1.2, on the 5th of March 2013. This new version was received as a real improvement, as will also become apparent in the following sections.

3.3 More Details about the Telehealthcare System

In this section I will describe the web-pages which were used for routine patient monitoring by COs and care providers, and the Excel spreadsheet which was used for documenting the patient state, on the latest version of the system (1.2).

The homepage (Fig. 3.1) contained the list of patients who were registered for the telehealthcare service, and allowed the user to select a patient (by clicking on her name) and access the pages which provided more details about her status. The Monitor page (Fig. 3.2) was normally only used by COs, as it highlighted in colour if the patient's readings had triggered an alert suggesting a possible worsening (exacerbation) in her condition. It also offered access, through a small circular button from its top right corner, to the patient's Patient Details page, informally called the 'badge'. The badge (Fig. 3.3) included demographic data on the patient to aid COs in the completion of the first page of the Excel spreadsheet. It could also include notes taken by other COs or by care providers about the patient (in the 'Obs' field). The Alerts page (Fig. 3.4) was the most important page of the website for COs, as it provided a table with the selected patient's current and previous alerts with their details, indicating which alerts were still open (needed addressing), and what their degree of criticality was (by means of red/orange highlighting). The user could open any alert by clicking on it from the table, and see its details displayed under the table. An opened alert could be edited to change any of its fields by clicking on 'Edit'. The Patient Monitoring and the Surveys pages were especially important for care providers, who used them in their assessment of the patient's state, while COs normally used them only to complete those questions

from the second page of the spreadsheet which had not already been addressed in Alerts. The Patient Monitoring page (Fig. 3.5) provided tables with the patient's readings, as highlighted in colour per type, and for CHF patients also as separated into type categories (as in the figure). It offered access to the Surveys page, which was not directly accessible from the homepage. The Surveys page (Fig. 3.6) provided a table with the patient's latest survey responses, with scores. The user could click on 'View' to open the survey and see all of its questions and patient answers.

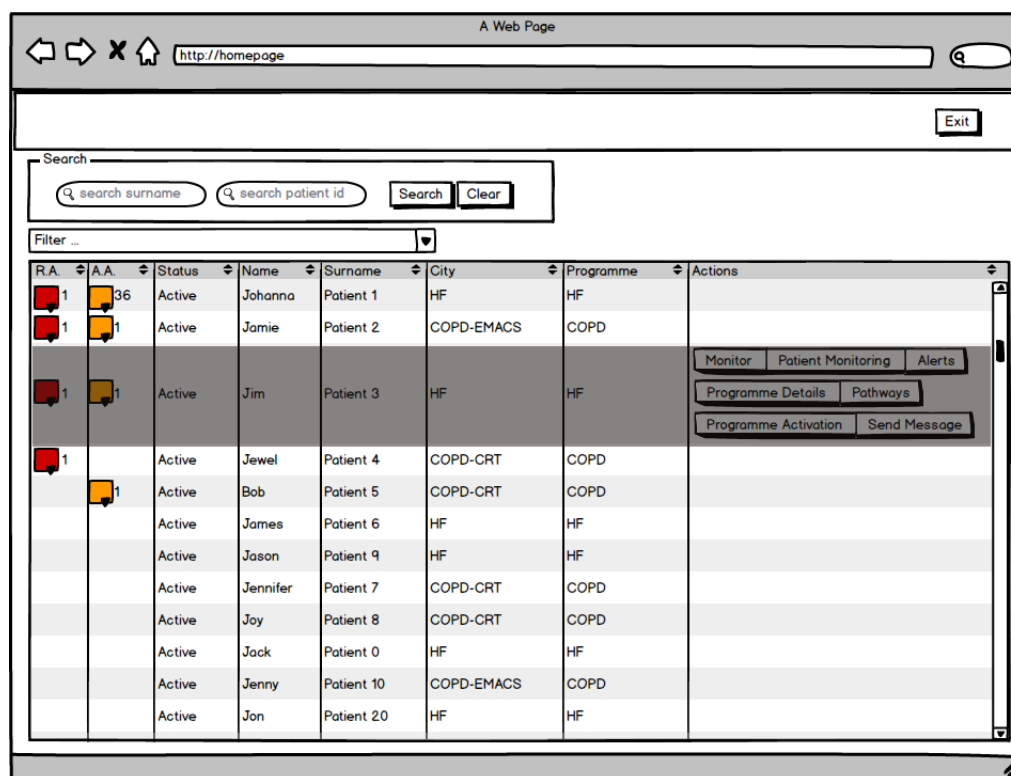


Figure 3.1: The homepage

On the first page of the Excel spreadsheet (Fig. 3.7), COs would fill in the details of the patient, as retrieved from her badge. Depending on the entered condition for the patient, the second page (Fig. 3.8) included appropriate questions (call back criteria) for the CO about the alerts which appeared for that patient on her Alerts page, and other questions about the patient state, the answer of which the COs could retrieve from the patient's other pages. The spreadsheet would guide the CO by a green arrow to the next question that they needed to reply to, and would suggest in its top white bar the next course of action. It could require the CO to telephone the patient, and guide her through a set of questions to ask her. Once all of the questions had been answered, or if telephoning the patient had not been required, the CO could enter her conclusion on

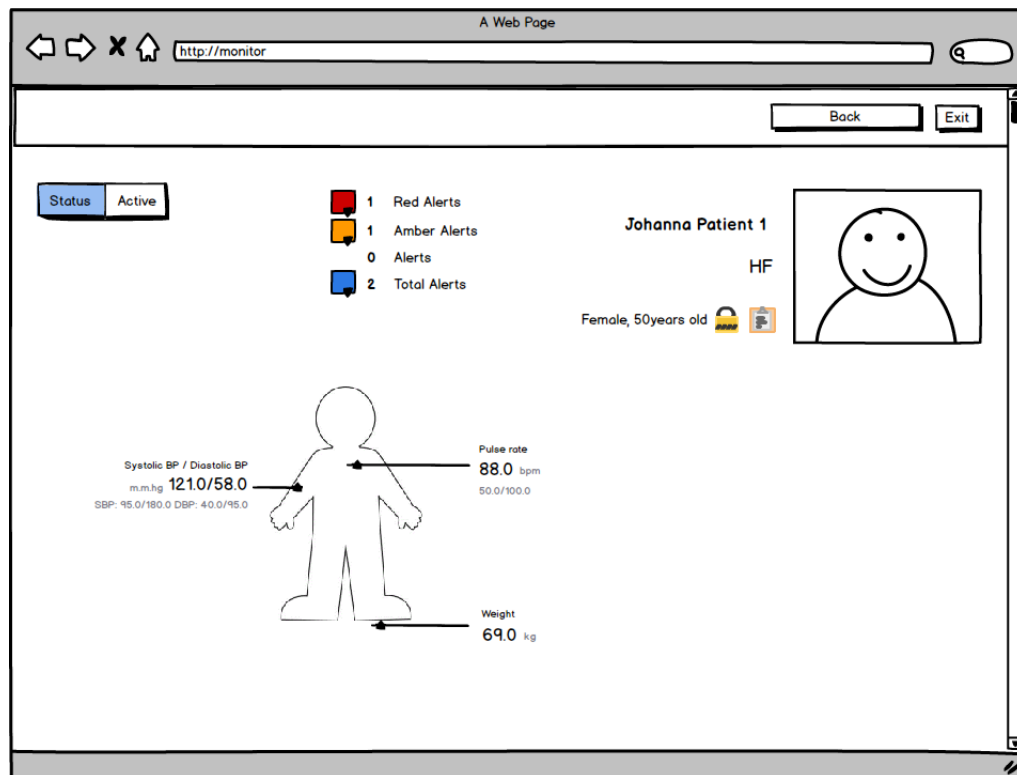


Figure 3.2: The Monitor page

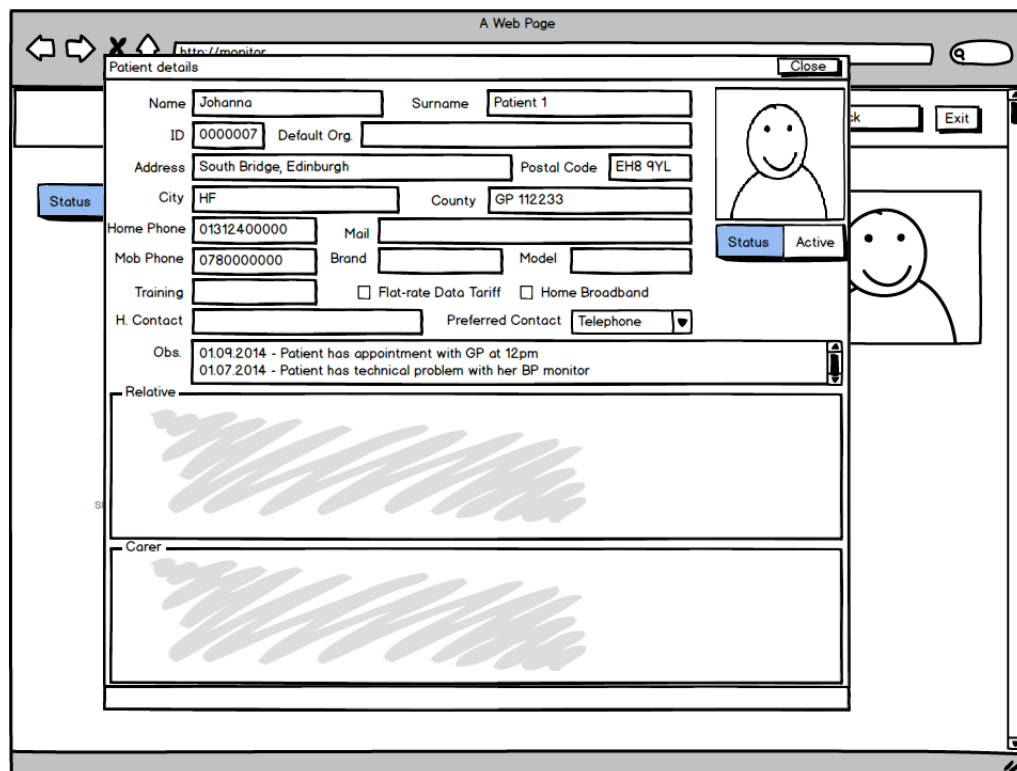


Figure 3.3: The badge

A Web Page

http://alerts

Alerts Johanna Patient 1 (01312400000) Back Exit

Search

Start Date End date Status Priority Search Clear

Date	Measurement Date	Measurement Time	Status	Priority	Description
28/07/2013	28/07/2013	09:09:09	Open	High	HF Total Change: Survey category increased
28/07/2013	28/07/2013	09:09:09	Open	High	HF Survey Score: Reading 2.0 is greater than ordinary
24/07/2013			Closed	Medium	Weight missing control: No received readings
24/07/2013			Closed	Medium	Blood pressure missing control: No received readings
17/07/2013			Closed	Medium	Blood pressure missing control: No received readings
17/07/2013			Closed	Medium	Weight missing control: No received readings
14/07/2013	14/07/2014	09:49:49	Closed	Medium	HF Total Change: Survey category increased

1 2 3

Details View history >

Status: Open Date alert generated: 28/07/2013

Description: HF Total Change: Survey category increased

Solution:

Comments:

Edit

Figure 3.4: The Alerts page

A Web Page

http://patient-monitoring

Back Exit

Weight On display

Weight

Blood pressure View

Diastolic BP

Systolic BP

Pulse Rate

View Surveys

Johanna Patient 1
Centre

Last records Display information in table mode Time: Last 7 days

Type	Date	Time	Value
Weight	28/07/2013	09:09:09	69.0 kg
Weight	17/07/2013	09:49:49	69.0 kg
Weight	12/07/2013	09:39:39	69.0 kg

Figure 3.5: The Patient Monitoring page

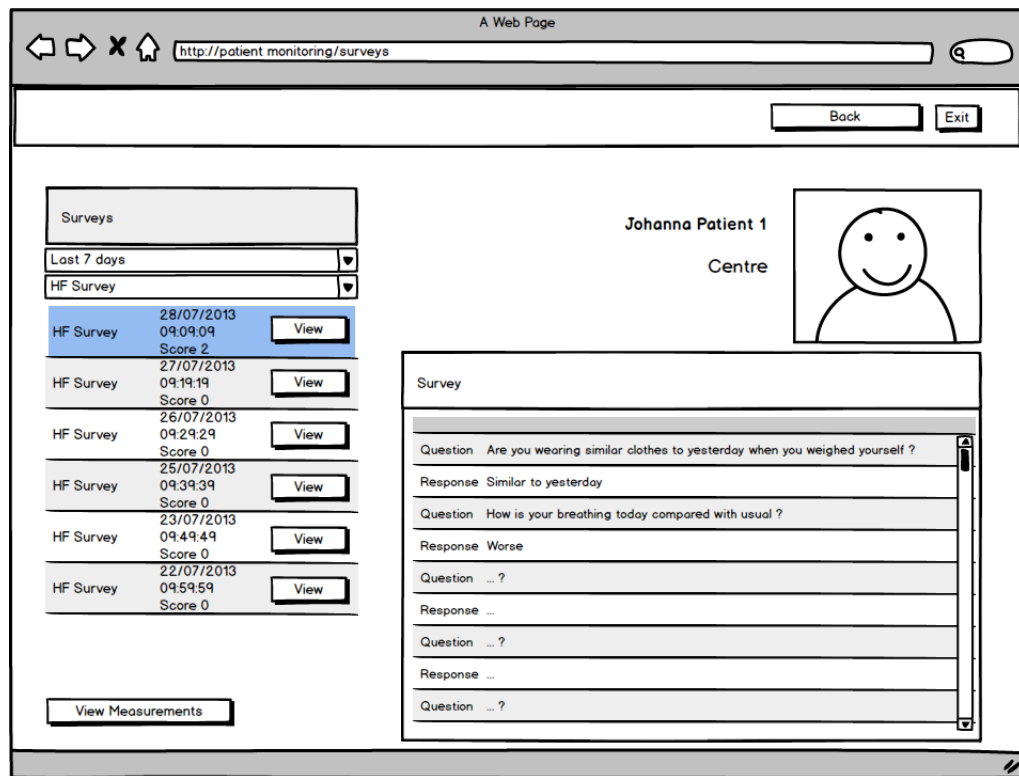


Figure 3.6: The Surveys page

the patient state, including the outcome suggested by the spreadsheet and her notes. COs would attach the completed spreadsheet to an email to the care providers, if the outcome required their notification.

3.4 Purposes of the Study

As a convention, I considered that the ‘system’ consisted of all of the four applications involved in patient home monitoring:

- **The telemonitoring website** which provided patient sent-in readings, answers to health-related surveys and triggered alerts to COs and care providers
- **The Excel spreadsheet** which the COs would fill in with findings on the patient’s state from the website and from their phone conversations, and which they would send, in case of a ‘Contact Care Provider’ outcome, as an email attachment to care providers. The care providers would use the Excel spreadsheets for being informed about the patient’s state and what had already been discussed with the patients.

Spreadsheet - Patient info page

ADVICE

Next step: Enter time (use shortcut)

Call operator

Operator name1

Guide

	Data field	Values
1.1	Date	28/06/2014
1.2	Time	
1.3	Patient name	
1.4	Patient ID	
1.5	Phone Number	
1.6	Care Provider	

CONTROL

PATIENT INFO

CHF ALERT

COPD ALERT

Figure 3.7: The first page of the Excel spreadsheet

- **The email facility** that COs would use to notify care providers about the patient's state and need for action
- **The statistics Excel spreadsheet** that COs needed to submit at the end of their shift to record the number of monitored patients, the types of alerts that they had triggered, the actions taken for each, the frequency and duration of calls, etc.

One of the main aims of my study was to gain an understanding of the different users' perceptions of the system's usability (the second aim is presented in Chapter 8). I planned to address the following questions: What are the usability problems of the system? In particular, what causes inefficiencies for the users in their work? What are the biggest problems, and what suggestions do users have? How were the changes brought to the latest version of the system perceived by the users, and how did they affect their work?

As described in sections 3.2 and 3.3, the users of the telehealthcare system needed to move in between several applications during their work on the system. In particular, COs needed to move information from the telemonitoring website to the Excel spreadsheet, to populate the email for the clinicians and to use the separate statistics Excel spreadsheet.

Spreadsheet - CHF Alert page

Patient name: Johanna Patient 1, Patient ID: 1, Date: 28/06/2014, Time: 12:00:01, Phone no: 07800000000

ADVICE Check score on system

Guide

	Data field	Values
3.0	Call back criteria	
3.1	Symptom score is 2 or more?	<input type="checkbox"/>
3.2.1	Increase of greater than 15kg over 2 days and an increase each day?	<input type="checkbox"/>
3.2.2	Weight greater than upper threshold?	<input type="checkbox"/>
...
4.0	CO calls Patient	
4.1.1	Was contact made with the patient?	Yes / No
4.1.2	Attempt 1	<input type="checkbox"/>
4.1.3	Attempt 2	<input type="checkbox"/>
4.2	Notes	
4.3	Absence of data alert - initial questions	
4.3a	Have you submitted a question today?	<input type="checkbox"/>
4.3b	Do you require any medical attention	<input type="checkbox"/>
...
6.0	Conclusion	Time
6.1	Outcome	<input type="checkbox"/>
6.2	Email sent to Care Provider	<input type="checkbox"/>
6.3	Notes	

CONTROL PATIENT INFO CHF ALERT COPD ALERT

Figure 3.8: The second page of the Excel spreadsheet

Clinicians needed to use the email coming from the COs, as well as the attached spreadsheet and the website to diagnose the patient. The lack of integration of these applications was a potential area of concern for the system, and I was requested to investigate its criticality.

Another potential problem was the fact that the telemonitoring website had just one interface for all of the user roles. This meant that users from each role had more information, web pages and functionality than they needed available. My usability study was also meant to elucidate whether the single interface was burdening users.

3.5 Participant Recruitment

Because they were all involved in the service and contributed to its evaluation, the system users from the different teams have all kindly agreed to participate to my study, in the limit of their availability. I made initial contact with their team leaders with the help of my study supervisor from NHS Lothian.

For the first part of the study, the NHS 24 team leaders kindly allowed me to visit their office every morning when a different CO was working on the system, which greatly facilitated my recruitment of COs. This part of the study commenced when the previous version of the system was still in use, and I recruited 4 COs at that point. The same users kindly agreed to participate again to a second part of the study after the new version of the system (Version 1.2) was introduced, in order to help me understand how this new version affected their work. I also recruited another 5 users- 3 COs and 2 team leaders for the study at that point, for the evaluation of the new version of the system. Overall, I managed to involve all of the COs apart from one person, who was not doing home monitoring at the given time, and their team leaders.

The second part of the study, the one involving the care provider teams, was a continuation of the study with NHS 24 staff, and only considered the new version of the system. I contacted participants by telephone or email, after getting permission from their team leaders. As the teams were small, I aimed to recruit as many of the team members for the study as possible. Overall, I managed to involve all but 4 care providers from the week-day telehealth monitoring teams, and one participant (a doctor) who was more experienced with the system from the LUCS weekend team.

I used an information sheet (available in Appendix A) to clarify the purposes of my study to NHS 24 staff and care providers, before getting their approval to participate. I also attached this information sheet to emails in the case of contacts made by email with care providers.

An overview of the total numbers of participants recruited for the study is provided in Table 3.1.

3.6 Data Collection Methods

I used a combination of **observations** and **semi-structured interviews**. In addition to these, during the first two weeks of the study, all of the NHS 24 COs working on the system at the time were asked to fill in a **diary** recording any problems that they

Team	Members	Study participants
NHS 24	10	9
CRT	7	7
EMACS	1	1
West Lothian Intensive Case Management Team	4	2
Heart Failure Nurses	4	2
LUCS	4	1

Table 3.1: Number of recruited participants per team

encountered on the system, in order to help me focus the interview questions. I then visited NHS 24 and observed two COs working on the system, without collecting any data, which helped me better understand the context before preparing the interview schedule.

The most important step of the data collection involved visits during which I conducted observations and interviews, with NHS 24 COs and care providers, and interviews only, with the NHS 24 team leaders or with care providers who did not have any work on the system that day (had not received any emails from the COs requesting them to monitor patients). I observed and interviewed each participant in turn. With the approval of the participants (given before commencing the observation by using the consent form provided in Appendix B), I video recorded the computer screen (only) on which they were working. During each observation, I also took written notes of the steps that the participants performed, the additional tools that they used, and if there were any arising issues. Observations lasted for the time that the person spent on the systems up to no more than 1 hour. However, care providers usually only had to monitor a few patients every day and spent less time on the computer. Observations were followed by half-an hour interviews with care providers and COs whom I was visiting for the first time, and fifteen-minute interviews with COs whom I had seen for the first version and was seeing for a second time, for the second version of the system. During the interviews, I asked participants questions about their experience with the system, their perception of its usability and their suggestions for improving it, but also questions arising from my observation (e.g. their motivation or contexts when using a certain functionality of the system or external tools). The interviews were semi-structured, which meant that, while having a structured list of questions prepared for them, I also followed interesting participant answers with other questions which

helped me elucidate new areas of interest for the study. The initial list of questions forming the interview schedules (available in Appendix C) were derived by me out of the purposes of the study, as discussed with my supervisor from NHS Lothian, and my previous experience with usability studies from my Masters. With the approval of the participants (also given in the consent form which I enclosed in Appendix B), I audio recorded each interview. I also made notes of interesting facts arising from the discussion.

Throughout the duration of the study, I also had access to a test website, to two patient tablets (one for a COPD and one for a CHF test patient) which I used to input data onto the test website, and to the spreadsheet to be filled in by COs.

Other data collection methods would have also been very beneficial for this usability study. In particular, it would have been useful to gather system logs of the users' interaction, containing information such as what web pages and functionality users accessed, how frequently and in what order, or how long they spent on each web page. As such logs can be collected over days and months, this would have provided me with much more data on the use of the system and what poses difficulties to the users compared to the data on these topics collected by me from the one-hour observations. Unfortunately, the company having developed the system did not have such interaction logs available, and I could not instrument the users' browsers with an application for collecting such logs due to NHS restrictions on installing new software on their machines. Observations were therefore my only option for collecting such data, while normally I would have used them only to understand contextual issues, what other tools users use, and the reasons behind their actions. I needed to keep observations short due to their disturbing and distracting effects on participants, which limited the amount of data that I could collect. Data logs would have also provided me with invaluable data about how the system works in practice. On the test website, with limited test data, I could not simulate all of the possible interactions, and I thus needed to also understand most of the system functionality, and identify problem areas, from the video recorded observations.

3.7 Data Analysis Methods

For both parts of the study, I watched the video recordings of the observations and made notes of the steps taken by the participants in their work, the tools that they used and any difficulties that they have encountered with the system. I compared the observations

made for different users, to better understand how they used the system differently and what motivated this. For each of the first 4 participating COs, I also compared observations for the two versions of the system, and with their instructions on using the system, to gain an understanding of how the new version of the system was used differently, and how this deviated from what they had been instructed to do in their training.

For both studies, I analysed the interviews qualitatively, by using the **template analysis** approach ([22]). This approach was found to be especially useful for applied research, as is the case for my usability study. It offers the following important advantages [22]:

- An increased efficiency of the coding effort, both by combining top-down coding and the very time-intensive bottom-up coding, and by building an initial template (i.e. hierarchical list of themes and subthemes) out of a subset of the transcripts only, to use for guiding the rest of the coding
- A consideration of the main questions to address in research, by allowing the use of a-priori themes. This is especially relevant for evaluation studies like my usability study.
- Flexibility, by allowing for the template to be modified if necessary in light of the remaining transcripts, or if the focus of the investigation changes
- A more focused investigation of the areas which are more relevant to the research question by allowing for unbounded levels of themes and subthemes (unlike other qualitative analysis approaches, which restrict this number), and the analysis of some themes in more detail than others

I started with some high-level a-priori themes derived from the major research areas which had been agreed with by my study supervisor, and around which my interviews had been centred. I then coded 6 interview transcripts out of the 13 for both of the parts of the study: 3 from users of each version of the system for the first part (involving NHS 24 COs), 2 from the CRT team and one from each of the other teams, for the second part (involving care providers). This helped me decide on an initial template consisting of themes and subthemes, one for each part of the study. This initial coding was done in two cycles, the first involving *open coding* with a combination of *descriptive*, *process* and *in-vivo coding*, and the second involving *code mapping* for reorganising the list

of themes and subthemes to reach more central ones (an overview of these coding approaches is available in [194]). After having decided on the initial template for each of the two parts of the study, I coded the remaining transcripts for each part by fitting citations into the initial template's themes and subthemes, only amending the template where absolutely necessary. In addition to template analysis, I also used *attribute coding* ([194]) for retrieving demographics data from the transcripts.

The full final lists of themes and subthemes, for the two parts of the study, are provided in Appendix D. In this chapter I will only describe the results for the themes which are relevant here.

My observations of the system from the videos, as well as my hands-on experience with the test website and spreadsheet, helped me understand the issues raised by the participants, and base my conclusions on the usability of the system.

3.8 Results

3.8.1 Demographics

User experience with the telehealth system varied from beginner to very experienced. For the first part of the study, the two NHS 24 team leaders whom I interviewed were very experienced with the system, having used it almost daily since it was first introduced. Out of the visited COs, two were beginners, having started using the system less than one month before the time of my visit, and only 3 times, respectively 6 times per total. Another CO was a beginner on the previous version of the system, having been involved in home monitoring by using it ever since the beginning, but with a very low frequency, but had more experience on the new version. Another CO was fairly experienced, having been involved since the beginning but normally working at most every fortnight. The other 3 COs were experienced, having been involved since the beginning and normally working every week, one of whom sometimes twice a week. While the team leaders had used the new version of the system almost every day when I visited them, the COs were all still beginners to the new functionality, having used it for 3-6 times only. They had a total of 4 hour working shifts monitoring patients using the telehealth system, but they sometimes spent less than this time, and considerably less since the new version of the software. Whereas previously they would have spent a whole shift (4 hours), many COs spent an average of 2.5 to 2.75 hours, and sometimes as little as 1.5 hours on the new version.

In the second part of the study, one participant had joined the CRT team 4 months before my interview. The other participants had been using the telehealth system since its introduction to their teams and had at least 6 months experience. Usage varied from fortnightly (2 participants) to weekly (11 participants), but it also depended on whether there were any emails coming in from NHS 24 about their patients alerting, and their number of patients. With the exception of 2 care providers who were very experienced with the system, having been involved in proposing improvements to it and to the service's protocols (one user from the Heart Failure Nurse team and the user from EMACS), it was therefore difficult to be fully aware of their levels of experience.

3.8.2 Support

All of the COs reported having participated as a group to formal training, during which two trainers had showed them how to use the system. One of the COs, as well as one of the team leaders, added that they had only been shown a test system during this training. Some COs also mentioned having monitored patients together with one of the team leaders during their first use of the system (3/7), or using role play to practice using the system (1/7). Other than the training, the great majority of COs also reported having available a folder with documentation on using the system, and IT contact documentation.

In what concerns care providers, most of the teams had been trained by the project's Clinical eHealth Advisor from NHS Lothian, who had visited their office and taken them through the steps of monitoring a patient on the test website, as well as demonstrated the use of the patient devices. One of the teams also mentioned that he had left them a tablet for being able to submit readings for a test patient and practice monitoring this patient on the test website. In the case of the Heart Failure Nurse team, he had only trained the lead nurse, who then in turn had trained her colleagues. The care providers from the CRT team reported having received in-house training from their team leader, who had demonstrated how to monitor a patient and had offered them some printed step-by-step information. Also, the participant who had joined the team at a later stage had shadowed her colleagues for two weeks after her training, before starting her work on the system. In terms of documentation, most of the care providers remembered having some available instructions on using the system. Also, a few participants mentioned receiving emails about updates to system changes. One of the members of the CRT team explained that they were sent by her team leader after she had participated to the

project's group meetings, which was often later than the changes being implemented in the system, and so she and her colleagues would need to cope with the changes in the meantime.

3.8.3 Navigation Problems

A system which is not designed with a clear understanding of its future users and their tasks may include design decisions which make navigation cumbersome, affecting efficiency. For example, if the designers do not foresee the users' routine need to access certain information from the system, they may not include any shortcuts in the system for easily retrieving that information and thus make it time consuming or even difficult for users to reach it. Therefore, navigation is central to our consideration of efficiency. In particular, I noticed that healthcare workers need to be able to do almost anything in a few clicks. I attribute this to their increased workload (having to manage a whole patient population as compared to interacting face-to-face with a much smaller number in traditional care) in the same amount of work time, which makes telehealthcare feasible only if it is efficient.

In this subsection we will look at the different problems that users had with their navigation through the telehealth system, the design elements which led to them, and the design suggestions that users made for improving navigation.

3.8.3.1 Slow Navigation Within the Website

3.8.3.1.1 General Comments

Most of the COs (5/7) considered that the system was a bit 'fiddly', only 2 COs judging the number of clicks required as acceptable (the two beginner COs). One of these participants mentioned getting used to the system, and another blamed the time it was taking her on her inexperience:

"you obviously still [do] a lot of flipping back and forth" (CO)

Most of the COs (5/7) mentioned that the new version of the system seemed to run quicker, and some of them explained that this was due to it running on the Google Chrome browser, as opposed to Internet Explorer for its first version. This was also mentioned by one of the team leaders:

"A lot quicker with the Chrome running it now" (CO)

Most of the care providers (8/13) considered that the telehealth system had important navigation problems leading to many numbers of clicks, which needed to be addressed:

“I think, you know, ehm... yes, it seems I just have to do an awful lot of sort of clicking on here, cause first of all, sort of going into the system, and then I have to click on here to make sure I’ve got all... all the CRT patients, sort of, here, and then I’ve got to find which patient I want, and then I’ve got to decide, you know, right ok-go into Patient Monitoring, click on that, and then back out of that and... you know, well, first on Alerts, then back out of that in order to find the Patient Monitoring system, and... yeah. It seems to be an awful lot of, you know, I’ve got to go down quite a few paths to get the information that I need.” (respiratory physiotherapist from CRT team, COPD)

3.8.3.1.2 Specific Navigation Problems and User Suggestions

The following were some more specific navigation problems mentioned by participants for the telehealth system:

3.8.3.1.2.1 Incomplete or Inappropriate Sorting Options

The telemonitoring website’s homepage contained by default a table with a limited number of patients, in scrollable form, and the rest could be accessed by using pagination. Patients were given in the order in which they had sent in readings for the day, and the table included all of the patients who were being looked after by the different care provider teams. There were options to sort the list of patients from the table alphabetically or by teams (the ‘City’ sorting option), but sorting patients by teams would not also render their list in alphabetical order. During my visits I noticed that, to search for one of their patients on the homepage, many of the care providers who were participants in my second part of the study (8/13) first sorted the list according to city and then scrolled down or even went to a different page to find the patient. Some of them explained that this posed navigation problems. Please note that, although few participants commented on the navigation required for searching for a patient, they and many other participants provided suggestions to address its problems, which made these problems real:

- One of the care providers who only scrolled to search for a patient (and was not aware of the alphabetical sorting option) explained that **searching for a patient**

was frustrating for her as she needed to do a lot of scrolling:

“That just annoys me, you do that every day, you think, somebody’s with their last name begins with an ‘A’ and you need to scroll down to the bottom of the second page to get it- ‘NO!’.” (respiratory physiotherapist from CRT, COPD)

This care provider as well as many other care providers (7/13) suggested correcting this issue by having the list of patients as presented alphabetically by default. I will return to the ordering problem and to the care provider’s suggestion from the perspective of presentation in subsection 3.8.4.2.1.

- Two of the care providers explained that having patients for all of the teams in the list and sorting them according to city led to **some teams always needing to scroll, and even go to a different page, to retrieve their patients:**

“Each time you go into it you have to click on ‘City’, and we’re lucky cause ours is at the top, but I am sure that the people that are these ones [belong to teams the names of which are placed on the bottom of the page] would get quite annoyed at having to go click on it and all the way down, ehm” (respiratory physiotherapist from CRT, COPD)

The two participants, as well as many others, suggested that the patient list only contain each team’s patients. I will return to this problem and participant suggestions in subsection 3.8.5, from a content perspective.

3.8.3.1.2.2 Information Inconsistencies, Compounded With Problem 1

Another problem with the list of patients from the homepage was that **some patients of the CRT and EMACS teams had been attributed to teams with incorrect names.** Two care providers explained why this problem made their navigation for searching for a patient difficult:

“The trouble is that when I search, at the moment, on City, I get COPD-EMACS, but then I have to go onto page 2, to get EMACS up as well.” (nurse from EMACS team, COPD)

3.8.3.1.2.3 Needed Functionality Being Difficult to Reach

A few care providers (4/13) and one CO explained that **saving information on the Programme Details page (e.g. new thresholds at which alerts were generated on**

the system, or key information about the patient having technical problems with her tablet or being in hospital) was frustrating. The new version of the system required users to pass through several screens until they reached the option to save updated information from the Programme Details page. The care providers explained that this was frustrating especially when they wanted to save only one piece of information. Some of them also presented this as a functionality problem, and leading to potential mistakes (not saving the information, mistakenly changing other information).

“I don’t see why you have to do that, it takes you through all the pages and at the end you can save it, but if you come out at any time it doesn’t save what you have done. (...) but actually you have to go through a course, whole course of pages to be able to change (...) it’s much more likely to cause errors because people won’t get to the end and then they’ll come out, so that nothing will be saved.” (CO)

However, one of the NHS 24 team leaders explained that in fact the key information which COs needed to change for the patient was saved by the system once a user clicked on ‘Continue’ to get from the first to the second page in the process. He added that only other information, to be saved onto the patient’s tablet, including the patient’s thresholds which care providers sometimes needed to change, required the passing through all of the pages.

I checked the functionality myself and, indeed, updated key information was saved when clicking on ‘Continue’ and returning to the homepage. There was a problem here with the button name (‘Continue’) not making it clear that it also saved the information, and leading the user to a different screen. Moreover, there was a warning message on the bottom of the screen: “Data changed. You must complete the process for the changes to be reflected in the patient’s terminal”. Despite explicitly mentioning that it was the patient’s terminal where the data would not be saved unless the process was completed, this message’s presence on the first screen, where one could only change information to be saved in the system’s database, was potentially very confusing. This problem therefore became a presentation and functionality one as well. Moreover, it continued to be an important navigation problem for the changing of the patient’s thresholds.

All of the care providers, as well as the CO, who commented on this issue suggested having intermediary ‘Save’ buttons at the bottom of each page, to allow them to save the information from that page for the patient:

“It just makes sense to have a ‘Save’, you know? Ehm, like so here in the front page, you should have ‘Continue’ or ‘Save’, so, on the very front page, if you just wanted to put in key information or put in somebody’s details, change a mobile number, you

should be able just to save it straight away.” (respiratory physiotherapist from CRT team, COPD)

Two care providers described another problem related to unnecessary navigation: **it was difficult to save into key information that a patient had been admitted or dispatched from hospital, as they needed to move from the Patient Activation page where they had activated/ deactivated the patient, and where they could only read the key information, to the Programme Details page for having access to the functionality of editing this information:**

“So you have to back into a different page to alter key information like the patients come in or come out or go into hospital. And again, it’s just frustrating, but again it’s just not... that’s the kind of clunkiness of it, that, you know, you can’t just click on the... the box that’s in front of you with the key information on, edit that, you have to go into a different page and edit that instead for it to come up on that so... again, for me, that’s just...” (respiratory physiotherapist from CRT team, COPD)

One of the participants suggested being allowed to edit the key information directly from the Patient Activation page.

3.8.3.1.2.4 Monitoring Work Information Not Being Available Where Required

As I also noticed in previous work ([25, 27]), healthcare workers need precision in the placing of information on the screen so that it is always easily available as needed. They do not have the time to look for it, and they seldom explore the system’s functionality other than that which that was shown to them in their training, especially if they have not had much experience with computers or with similar systems in the past.

Once on one of the pages presenting information about the patient, the user could only move to another of the patient’s pages by clicking on the website’s ‘Back’ button to return to the homepage and then clicking on the patient’s button corresponding to the wanted page. This was seen as negative by the great majority of the care providers who offered comments on navigation (8/10), especially in what concerned the movement between the Alerts and the Patient Monitoring pages:

“One of the things that’s maybe slightly annoying is, when you go into... the Alert, and you want to look back over the last 7 days, you’ve got to come out of that alert, and go back into another page to see what the trend has been.” (nurse from Heart Failure Nurse Team, CHF)

One of the care providers explained that moving between the Alerts and Patient Monitoring pages was necessary for understanding the patient's state and making a decision on it, as the Alerts page did not offer enough information:

"[referring to alerts] So all that that gives you there is information on... that it's over threshold, doesn't give you any information about what's happened over the last week. And that's really the important part, so you have to come out of there, back into Patient Monitoring, and have a look." (nurse from Heart Failure Nurse Team, CHF)

The care provider also explained that this was especially problematic if she was on the phone with the patient, or if she was looking for a colleague's patient. She mentioned that she used to make a note of the alert such that she would not forget it when going to see the patient's readings.

Care providers offered two classes of suggestions for this problem:

- **Being able to move in between the different patient pages once within the patient**, offered by the majority of the care providers (6/8) and agreed on as suggested by me by another care provider:

"It would be nice if you could go, within her screen you could move around, from Patient Monitoring to Alerts, you know?" (doctor from LUCS team, COPD and CHF)

Most of these participants (6/7) suggested that links exist between both of the pages, such that one could move freely between them. Although these comments mainly concerned the movement between the Alerts and Patient Monitoring pages, most of them (5/6) generalised this, suggesting that links exist between all of the patient pages.

- **Joining the information on alerts and readings**. One care provider suggested having alerts next to readings in Patient Monitoring and being able to click on a button to close off the alerts:

"I went into Patient Monitoring, and that's the weight, if the alert was beside the weight... he would... and again, that's what [the previous system] I think had that... so if you had the alert there, the alert there, the alert there, and you wanted to close off the alert, you could just press that alert then, and close off the alert." (nurse from Heart Failure Nurse Team, CHF)

Another care provider suggested having signs and symbols marked next to the representation of the readings (for weight- in which days it alerted and by how

much it has increased), or being able to click on alerts to see the last 7 days of readings.

During the first study, one CO also commented about the cumbersome and time consuming navigation from the Alerts page to the Surveys page, by returning to the homepage, going to the Patient Monitoring page and then from it to the Surveys page which was only accessible from there. She suggested there to be a direct link from Alerts to Surveys:

“(...) obviously if the alert or a couple of alerts and you’ve got, say, their bp [blood pressure] is outwith the threshold, that’s red, so you know that’s one alert. Erm... but you don’t have the other alerts or surveys in here, so maybe it would be a good idea to have the survey, you could just click it straight into survey from that page. (...) so yeah it would probably be a good idea to be able to get into that from that page. Rather than going back, going into Patient Monitoring, to then go into Surveys... so it’s a lot of clicking about for people.” (CO)

The following were other navigation problems when retrieving monitoring work information identified for the system:

- Out of the pages which were used by care providers, **the full list of patient reading thresholds could only be retrieved and changed from the website from the second screen of the Programme Details page**. Three care providers explained that it was important to see the thresholds for a patient because they needed to change them often, and because they helped them understand how far from the thresholds the patients’ readings were. One of these participants also explained that she and the out-of hours team would need to see the thresholds because they did not share this information any other way. What was clear was that care providers needed thresholds to be more readily available to them and to avoid navigation for reaching this information.

Two care providers proposed having the thresholds available next to readings on the Patient Monitoring page, and as suggested by me another care provider agreed. Yet another care provider suggested having the thresholds available on a separate page and accessible through a ‘Thresholds’ button from the homepage:

“I personally feel it would be really good to have a separate ‘Threshold’ button, because that is actually what I tend to go into the most, and particularly if you have a new patient who goes onto telehealth, you’re playing with their thresholds

all the time while you're learning what's normal for them.” (nurse from EMACS team, COPD)

COs would normally write threshold values in their outcome on the Excel spreadsheet, and one of the COs also explained that she found it cumbersome to retrieve thresholds from within the Programme Details page. I will return to this point in subsection 3.8.4, when describing content problems.

- The responses that patients provided in their health-related survey were available to the users on the Surveys page. One care provider commented on the fact that **one needed to open each survey separately**, and another care provider offered suggestions in this respect which made it clear that **it was important for them to be able to compare survey responses from consecutive days**. Both of the participants suggested having survey answers which have changed as highlighted on each survey.

I will return to these points in subsections 3.8.4.1.1 and 3.8.4.2.2, from the perspective of presentation.

- Two care providers commented on the fact that **one needed to open up all of the alerts in order to understand what had happened with the patient over the last days**:

“If I’ve been on holiday and I come back in and I say ‘wonder what’s been happening with that patient’, you still have to go in every day...to find out.” (nurse from Heart Failure Nurse Team, CHF)

One of the care providers suggested having a ‘History’ button where one could access the last week’s comments on alerts, or having the alerts open by default. When I proposed the second solution to the other care provider, she also agreed that it would be useful.

“Me: So what would you prefer, what would you envisage as a... ”

Participant: That should say ‘View history’ and maybe give you a week’s history of... of these comments. (...) [referring to alerts] Or even if that [the history] was, say, open like that all the time... ” (nurse from Heart Failure Nurse Team, CHF)

- Patient readings were recorded on the Monitor page (last readings only, as highlighted in red if they exceed thresholds), and Patient Monitoring page (the history

of readings, not highlighted). COs normally preferred to see the highlighted readings from the Monitor page, but two of them explained in the interview that this sometimes caused them difficulties when asking patients to retake their measurements, because they would need to refresh the Monitor page or move in between pages (Monitor and Patient Monitoring) to realise when the new readings were received and to retrieve old and new readings. They usually preferred in this case to write the two sets of readings down:

“Other times that I personally take down details is when I am comparing their SATs [oxygen saturation], so if I’ve asked them to repeat their readings I’ll write it on a bit of paper, say it was, say, 82% for oxygen, 102 for pulse, I’ll put an arrow to mark down on a bit of paper what I’m going to put down into the next box, just so that you are not constantly flipping back from screens and sometimes if you refresh it, you would have to flip back and just double check with their old readings so I just write it down so I can see the difference.” (CO)

- COs needed to reply to a series of questions from the spreadsheet regarding the existence of different alerts for the patient, and thus be guided to an appropriate course of action. Despite being trained to reply ‘no’ to questions which did not have an associated alert, there were some cases where they needed to also go into other pages than the Alerts page to be able to answer the question. In particular, two COs mentioned **having to go into Patient Monitoring- Surveys in order to answer the question from the spreadsheet regarding the increase in survey score, which they found as inconvenient:**

“(. . .) it’s just sometimes if you look at the survey scores, the ones that have alerted when you’re in the Alerts page, and then you have to go back and see, erm, if it was more or less [according to the question from the spreadsheet] you would then you would go into your . . . Patient Monitor, and check your survey scores for that day and go back into that [Alerts].” (CO)

When asked by me, both of the users agreed that it would be useful to have the previous day’s survey score mentioned in Alerts as well:

*“Me: Would it be helpful to have those things highlighted in alerts as well?
Participant: Probably, cause if you’ve seen that it’s highlighted at a certain point, then ‘yesterday’s was’ . . . it would be more useful. ” (CO)*

As I noticed from the interviews, there were also other questions from the spread-

sheet for which COs always needed to check other pages than the Alerts page, this leading to additional navigation. For example, the spreadsheet included a question about the patient retaking her measurements, for which COs needed to check the Patient Monitoring page. Also, it included a question about whether there was a reason in the patient's notes as to why a patient had not taken her readings, for which they needed to check the notes from the badge accessible from the Monitor page.

- **In the first version of the system, the readings from the Patient Monitoring page were provided in a graphical representation by default, but as I found both in my observations and from interviews, COs always used the table mode, and the majority of the care providers (9/13) preferred it. In Version 1.2 of the system, as was requested by the teams, the table representation was made the default visualisation. More than half of the COs (4/7), and one care provider, explained that this was an improvement as it saved them one step:**

“(...) changed to table mode, cause that's the one we use, so it's... it saves a step from having to change from graph to table mode, it's better to just go straight into... to table mode.” (NHS 24 team leader)

3.8.3.1.2.5 Problem 4 Compounded With Problem 1

Added to the problem described in the previous section, the time it took to move between the different patient pages was also negatively influenced by the fact that, **on going back (by pressing the 'Back' button) to the homepage, users found the list of patients which they had previously sorted as unsorted and they needed to sort it and search for the patient again. This problem became even more critical if the patient was not on the first sorted page. In this case, when going back users found the page on which the patient was normally placed as unsorted, and the patient not there any longer, and they needed to return to page 1 to resort the patient list, and back again to that page to find the patient.** This problem was also noted by the majority of care providers who made comments on navigation (7/10):

“(...) sometimes you're in Alert and I think 'well, I wonder what that showed yesterday' [from the Patient Monitoring page], so you go BACK and then you're scrolling ALL the way down again... (...) And if you're on page 2, you then go into

all of that [going to page 1, sorting, going back to page 2].” (doctor from LUCS team, COPD and CHF)

The majority of the care providers who explained this problem proposed being able to move between the patient pages once within the patient from above, which would make this problem no longer an issue. Before making this statement, 3 care providers also suggested that the sorting problem be amended:

“When I want to go back, I want to go straight to that page, well than have to click on ‘City’ to get all the CRT list again, so that would just speed things up and it wouldn’t be so staccato, you know, do something and then have to jump, do something else, rather than just being seamless.” (respiratory physiotherapist from CRT, COPD)

3.8.3.2 Slow Navigation Between System Applications Due to a Lack of Integration

3.8.3.2.1 General Comments

More than half of the COs (4/7) and some care providers (5/13) mentioned that the fact that they needed to move in between three applications (system-spreadsheet-email facility) for monitoring patients was confusing and time consuming:

“In this one you’re kind of clicking back and forward looking at things, and attach the emails, and... so slightly fiddly this one. (...) you are asked something ... you’re doing this form, it will ask you some questions, you got to go back on this one and check...” (CO)

“To get the email and then open up the spreadsheet, and then go into the website, it’s like two steps too many, you’re as well looking at the website” (nurse from Heart Failure Nurse Team, CHF)

One of the care providers also explained that this was made worse by the fact that her email client required her to log in again after a certain timeframe:

“There’s two screens, so you’re logging into two screens. Now, because they can take quite a while to come through, I’d done the two [patients] that are there today, ehm, but by the time supposing... you know, another twenty minutes another one comes in, the first screen will have... cut me off, because I’ve been waiting, so you have to log back into it again, so it is more cumbersome” (doctor from LUCS team, COPD and CHF)

Both during the interviews, as well as during some observations, I learnt about different strategies that users deployed for dealing with the movement between the three

applications more efficiently:

- Two of the COs mentioned that they would write the patients' details down on paper to avoid moving between the email and the system looking for them:

"Well, at the end of a... each phone call, we have to email the care provider with an outcome. And in that we have to include the patient's name, telephone number, which care provider it's to go to, so I write it down just to save me having to go back and find the information again, it's just for time, really." (CO)

- Two of the COs explained that they would write outcomes on paper for statistics such that they did not need to repeatedly go in between applications. I also observed more than half of the COs doing this:

"[referring to outcomes for statistics] rather than go into another computer screen, I'm quite happy to write it down." (CO)

- I observed some COs writing alerts down before filling them onto the Excel spreadsheet.

- Two care providers mentioned writing patient names down from the email and then checking all of them on the website, one of them in relation with her comment from above about the email client requesting her to log in again after some time:

"If you come in [the email account] and there's 6 names there, then it's not practical to go in and out of the website 6 times, it's better for me to just go through the most of the 6 [from the emails], write down anything wrong, and then look up on... on the system, and then check them all off." (respiratory physiotherapist from CRT team, COPD)

3.8.3.2.2 Suggestions

When asked about the potential to somehow integrate the different applications that they were using as part of the system (Excel spreadsheet, website, email facility and statistics table), all of the COs agreed that it would be useful for them, and one of them even proposed it herself:

"it would be nice if it was more streamlined (...) you know how you've got the telehealth system of the patients, and then you've got your email system, and then you've got your form, your home monitoring form, it would be good if you could just have that onto one system" (CO)

One of the NHS 24 team leaders explained that the setting up of the system was error prone and time consuming, which motivated the integration of the applications:

“All of the transfer of demographics and information, you know, it’s where the errors come up because they [COs] are transferring so much information and they’re slipping a finger and they’re clicking ‘yes’ and you’re meant to click ‘no’, and that throws the whole thing off, so if that was all done for them, then they could just really focus on. . . on the call side of it, on the patient side of it, rather than having to fill it with paper aside, which I think. . . if we look at call times, it’s probably what. . . most of the time is, you know, 10 minutes an alert, you’re looking 4 minutes. . . 3-4 minutes for a call, so you can have 5. . . at least half that call is paperwork, essentially.” (NHS 24 team leader)

One of the care providers understood integration as making the movement between the applications more seamless and proposed the existence of a link from the spreadsheet to the website. More than half of the other care providers (7/12) agreed that complete integration would be a good idea, two of them proposing it themselves in response to their criticism about the time it took to move between the applications. Another two participants did not rule it out but were not sure about it as they could not envisage the change it would bring.

The care providers who rejected and one of the care providers who was not sure about the idea of integration (4/12) explained that the email was useful for them to be able to decide whether to go onto the system. Two of the care providers worked as part of the West Lothian Intensive Case Management Team. One of them explained that, as they worked within a shared group, she would like the system to continue to include emails as they helped them share responsibilities:

“I guess the only thing that the email is. . . it tells you straight away that. . . there is somebody [a patient] there. So, if you’ve got a busy day. . . it’s probably, for me, less chance of you maybe missing somebody, because you don’t have the, sort of, the numbers to look through. (. . .) For me, I think I would prefer the system that we already have, ehm, and I’m just wondering how. . . because we work, sort of, on a. . . a sort of shared group there, it could be 3 or 4 of us, so. . . I’m trying to think how that would maybe work as a sort of team as well.” (nurse from Heart Failure Nurse Team, CHF)

The other care provider motivated the use of emails for being able to also check them when out-of-hours or in their other office.

Another care provider considered that integration was one option, while another was for NHS 24 to send them emails containing more information, which would prevent

them from going onto the system for the patients whom they were looking after:

“If the... email attachment or the email alert came through from the... telehealth system, with some information, it might prevent you going into the telehealth system because... you know the patients, if you’re... if it’s your patient that the alert’s about, eh... so you may not need to log into the system because you might have enough information generated from telehealth, and I think if you had enough information coming... if I came in and I logged in and I got an email and, it was like, oh there’s 3 emails from [the system] about A, B, C and D, alright oh that says that, alright.” (nurse from Heart Failure Nurse Team, CHF)

The following are more specific comments that I received regarding the integration between the different pairs of applications:

3.8.3.2.2.1 Website-Spreadsheet Integration for COs

NHS 24 COs were using the Excel spreadsheet for each patient, to record the patient’s alerts and be guided by it to a course of action. Following the suggestions of the spreadsheet, they would call the patient and record her answers and the summary of the phone conversation in it, and/or contact the care provider team and attach it to the email.

In the first part of the study, all of the participants (including the two team leaders) recognised that the integration between the website and the Excel spreadsheet was desirable, although some of them (4/9) could not imagine how this would be achieved. Two of the participants who offered suggestions in this respect proposed as a solution the inclusion of the spreadsheet within the website, in which case they would still need to fill it in.

One of the team leaders compared the system with another one that they were using, and suggested that the patient demographics be pre-filled in the spreadsheet, the COs just needing to fill in the alerts:

“If I went into a page, well I... the patient triggered an alert, then I can go into my system and say ‘right, all that’s done’ [the patient demographics], I don’t have to fill out anything, because all the information’s there, all I’ll gonna have to do is put in what they’ve triggered as alert, and then I can run through the call... so, like, the spreadsheet... the spreadsheet tabs on the system” (NHS 24 team leader)

When suggested by me, she agreed that it would be ideal to also have the alerts filled in, but expressed concern as to the efficiency of the system:

“(...) how complex would it be to open this record and have everything filled out? (...) In a perfect world, it would be patient demographics, alerts filled out.” (NHS 24 team leader)

Other 2 COs considered that the solution of the system automatically filling in the alert information would be very useful:

“Me: Would you like some of the information to be automatically filled in there when you do this?”

Participant: Yeah, that would be helpful, wouldn't it? Yeah...

Me: Because I noticed that you need to select the answers for the alerts...

Participant: Yes, from the Alert page, when you're putting over, yes that would be great, aha, it would save a lot of time.” (CO)

3.8.3.2.2.2 Website-Spreadsheet Integration for Care Providers

Care providers were receiving Excel spreadsheets as attachments to the emails from NHS 24 COs. They would use them to understand summary information on the patient's state and what had been discussed with the patient, and thus reduce the need to look into the patient's alerts, sent in readings and survey results from the website.

In the second part of the study, out of the care providers who provided comments on the integration between the three applications (6/12), some care providers (3/6) explicitly commented on the use of the spreadsheet when integrating the applications. Two of them suggested (one in reply to her criticism about the navigation between the applications) that the spreadsheet could be available on the homepage from a link for the alerting patient:

“Would you be able to do an email link there [on the homepage, next to the patient's name], the NHS 24 email link comes up in that (...) so the email would be there and that would have that Word template.” (respiratory physiotherapist from CRT team, COPD)

The third care provider suggested not using the Excel spreadsheet any longer, as she did not find it useful. Instead, she recommended adding a better notes facility (instead of the key information textarea from the telemonitoring website) where COs could also record their conclusions:

“I think ditch the xls spreadsheet, because to be honest it doesn't really tell me anything, because... it's, it's... I think the, sometimes the quality of the information at the bottom of the xll... xls spreadsheet, is not good enough to make any sort of

clinical decision, so it might as well... might as well not be there, ehm, and the other information that's provided you can find on the system." (respiratory physiotherapist from CRT team, COPD)

"The notes pages that are kept, so the notes that you add on at the moment, I think that could be better, and I think that could be integrated with what NHS 24 want their Excel spreadsheet to be as well. (...) I don't know if like a box that opens on the screen, just a notes box, would be more... helpful, rather than clicking on to seeing a line of information to the side." (respiratory physiotherapist from CRT team, COPD)

3.8.3.2.2.3 Website-Email Facility Integration for COs

NHS 24 COs were using the email facility to contact care providers regarding patients who were requiring further attention (as suggested by the outcome of the spreadsheet).

More than half of the participants to the first part of the study, including the two team leaders (5/9) proposed or agreed to my suggestion of an integration between the website and an email facility. They all suggested the functionality of automatically generating the email for the care providers from within the website. One of the COs who proposed this functionality, and the two team leaders, also suggested that the email be pre-populated with patient information:

"I guess it's technology again, but you could have it that it automatically sends the email rather than us, cause we fill out the form... but to fill out the form it tells you then to select from the email, but if there's pointers within that form it could probably generate the email for us... less room for human error" (CO)

3.8.3.2.2.4 Integration Between the Email Facility and the Telemonitoring Website for Care Providers

Care providers were using the email facility to receive emails from NHS 24 COs regarding patients who needed their attention, thus to be informed about the patients whom they needed to check that day.

6/7 of the care providers offering comments on full integration, who commented on this subsection, suggested that the website could alert them as to the patients that they need to check. Some care providers suggested that the alerts could be represented by highlighting the patients in one or more colours (3/6) or as an 'Alert' icon (2/6):

“But on the, ehm, [system] screen, if the alerts came up on the patient, so... if you went into the [system] screen and it had- ‘Alert’, or red, or something, you know that you would then know to go into those patients.” (doctor from LUCS team, COPD and CHF)

Also, some of the care providers (3/6) suggested still having access to the text of the email in order to retrieve its details:

“I don’t know, highlight the patients in red who need an action, and then you click on them, they would send, essentially what they send in the email, saying, you know, this is ‘for information only’, this ‘needs action’, ‘this patient hasn’t transmitted’, so that kind of thing.” (respiratory physiotherapist from CRT team, COPD)

“There might be something along here, that said ‘Alert’, that you could click on and it would tell you what the alert is, and then as you come out of that, you can just go and see.” (respiratory physiotherapist from CRT team, COPD)

Another care provider suggested that the information from NHS 24 could be reflected by the colour of the alerts:

“They sometimes they say ‘for your information’ and sometimes they say ‘for your action’. So amber could be for your information and red could be for your action, and it would go down... so, immediately you would know, trrrrrd, ok, I’ve got three ambers or five red, I’m going to do my reds first, then I’ll have a look at the ambers, and then I’ll go from there.” (respiratory physiotherapist from CRT team, COPD)

Some of the care providers (3/6) also commented on the importance for their navigation that the alerting patients be placed at the top of the patient list:

“If you had alerting in the website, if it was done in... well, say you had 100 patients, and you had 10 alerts, then these 10 alerts would be at the top, so then you don’t have to scroll through 4 pages... to find the alerts.” (nurse from Heart Failure Nurse Team, CHF)

One of them also added that it would be enough if she just had access to the alerting patients, while the others could be hidden and only accessible on request.

Apart from the 6 participants offering comments on the full integration of the email and telemonitoring website, and although not mentioning being bothered by the navigation between the different applications, another 2 care providers suggested that having a link between the alert email coming from NHS 24 and the telemonitoring website would make navigation more streamlined:

“I guess, you could... if you could... ehm... go straight from... the, the email... and click on it and that would lead you straight to your clinical management portal, instead

of having to then... separately access it, that would be... that would be very streamlined.”
(nurse from West Lothian Intensive Case Management Team, COPD)

3.8.3.2.2.5 Website-Statistics Spreadsheet Integration for COs

COs recorded at the end of their shift statistical information (e.g. number of patients, outcome for each, duration of calls, etc.) into a separate Excel spreadsheet. One CO mentioned having proposed to her manager the integration of the website and statistics spreadsheet, as motivated by the desire to have more accurate statistics, and, when prompted by me, other 3 COs agreed with this suggestion:

“I did say to [the team leader] one day the other day, would there be any way that they would be able to count the alerts that we’ve worked that day and whether we have signed into ‘care provider for action’ or ‘no action’, because, you know, we are taking a list of these, would it be able to count them and what we are doing with it? Yeah, I think that would be quite a good idea, and it would also get accurate statistics. There may be times when two of us are on and we don’t always get accurate statistics to be honest.” (CO)

3.8.4 Presentation Problems

In any software system, the structuring and representation of information according to user needs makes it easier for users to understand and use it in their work, while the opposite may cause confusion and frustration, and make users less productive and more prone to making mistakes. Given the safety-critical nature of telehealthcare systems, and the importance of efficiency for monitoring a higher number of patients, it is thus especially important for healthcare workers using such systems to be presented with information which is understandable at a glance.

3.8.4.1 Problems With the Layout of the Website Pages

The layout of a website page has to do with the whole organisation of its visual elements, including its headers/footers, tables, buttons, form elements, graphics, etc. From my observations from previous work ([25, 27]), I learnt that healthcare workers need a compromise between the placing of related information on the same screen, and cluttering the screen too much. In particular, cluttered screens decrease the users’ capacity of finding information and following ideas, and their productivity in their work.

Productivity is especially important for telehealthcare systems.

The following are the most important page layout problems which were identified by the users for the telehealth system:

3.8.4.1.1 Needed Information Only Being Reachable by Using Scroll Bars

This problem was reported for several areas of the system:

- **On the Surveys page, where users needed to scroll down to retrieve the answers that the patient had given to the last two questions from the health-related survey.** This was reported as a problem by several care providers (4/13). One of them also mentioned her difficulty of comparing a patient's last two answers from the last two days in this circumstance, as she needed to scroll down in the first survey, open the second and scroll down into it as well:

“Another sort of small bug there is the fact, when you, when you click on and off, it doesn't actually show you this whole screen, so you... it's not... comparing the two days, you still have to move down, do you know?” (nurse from West Lothian Intensive Case Management Team, COPD)

Two of the participants also explained that the last questions from the surveys, which were thus hidden and one needed to scroll for them, were also the most important ones for their assessment of the patients' state:

“So, it's only a few, and then the most important ones too, if you're taking antibiotics and steroids, so you know... one of... one of the important things is that, are they taking antibiotics and steroids, you know, and if they are, then you know they've got chest rates, you're not so bothered about it, so they'd manage, but...” (respiratory physiotherapist from CRT team, COPD)

They all suggested improving the layout of the page for fitting all of the survey questions and answers in.

- **On the Alerts page, alerts triggered for the patient were provided in the table with date and time, a coloured band highlighting them and a description, but to read the full description one needed to scroll to the right.** A few care providers (3/13) criticised this issue. One of them mentioned this in relation to retrieving information on the patients' thresholds, which was available within

the details of alerts concerning readings only if scrolling, and was thus somewhat hidden:

“[referring to alerts] It’s already written on there, ehm, but you don’t see it till there, and because you have to scroll across I don’t think many people do” (doctor from LUCS team, COPD and CHF)

Another care provider considered that the table of alerts contained quite a lot of white space, and that the coloured bands did not necessarily have to be that long, and therefore that the alert details could be made visible without scrolling:

“Participant: there seems to be quite a lot of wasted space between. . . bits here, and actually you end up having to read it, to go all the way along, yeah.

Me: To scroll.

Participant: Yeah.” (respiratory physiotherapist from CRT team, COPD)

“It would be useful if we didn’t. . . again, if it could all be a bit more compact (. . .) and actually, how big does that box need to be, because, you see, the red or yellow, whether that needs to be that length or that length, it doesn’t matter, so I think it could be more compact until it actually fits on a page, ehm, so it’s easier to read it, ehm. . . ” (respiratory physiotherapist from CRT team, COPD)

Two participants from the first part of the study also suggested that the scroll bar could be removed for seeing the full details of the alerts right from the start:

“ If they made measurement date, measurement time and status a bit shorter, it would probably pull it off, but it’s not, it’s not something detrimental” (NHS 24 team leader)

3.8.4.1.2 Use of a Fixed Width Layout

During my visits, I observed that **some care providers preferred to keep their browser window smaller** to be able to move in between applications easier. In this case, the website was not adjusting the size of the patient table, but adding horizontal scroll bars. One care provider, who normally searched for a patient by scrolling, described just how **difficult and time consuming (efficiency problem) it was for her to do this** in this context. This problem is related to the navigation problems for searching for a patient described in subsection 3.8.3.1.2.1, and with the problem of not having patients as ordered alphabetically on the homepage by default, as described in subsection 3.8.4.2.1:

“You’re going up and down this list, you get to here, and you can’t go any further, it stops, so you then you have to scoot along, then you have to go down there, and then you have to go into page 2, so, I mean, I would guess that’s got to be quite easily sorted, but it is a REAL PAIN, you know?” (doctor from LUCS team, COPD and CHF)

“You don’t get it all in, do you see what I mean? I need to scroll over.” (doctor from LUCS team, COPD and CHF)

“It’s just. . . if they’re gonna get more and more and more patients, then you’ll be on page 3 and 4 and. . . I take it 5, cause I think what they want at the end of the day is a central. . . somebody, looking at the whole of Scotland or something like that, but, I mean, you could be screening down hundreds of people.” (doctor from LUCS team, COPD and CHF)

3.8.4.1.3 Buttons Only Being Accessible by Using Scroll Bars/ The Website Requiring a Higher Resolution than the Users Have Available on their Screens

Although the following comment was only made by two participants, a CO and a care provider, it is interesting in that it emphasises the importance for functionality to be available at the users’ fingertips, otherwise causing inefficiency and frustration.

The Patient Monitoring page, containing the patient’s history of readings, had a button on the bottom left hand side leading users to the Surveys page, where the patient’s history of answers to health-related surveys was presented. One CO mentioned having had difficulties with finding the ‘View surveys’ button from the Patient Monitoring page when she had first interacted with the system:

“ And to be honest when I first started it took me forever to remember how to get to their survey, because it is right down at the left hand side (. . .) the call operator has to scroll down to find it, so if you’re new (. . .) you don’t know where to find it, it’s quite hard to find it.” (CO)

The participant mentioned needing to scroll to find the button, which may be explained by the resolution of her screen. Indeed, from observations, I noticed that COs worked with HP L1950g monitors, the native resolution of which is 1280 x 1024, lower than that which the website seemed to be requiring. Moreover, my observations confirmed that, depending on the resolution set up on the monitors, the buttons from the bottom of the web pages were not always visible to the users and that they sometimes needed to scroll top-down, left-right within the pages to retrieve all of the information.

One of the care providers, who used a smaller window size in her work, also told

me that she had found it hard to realise that the Patient Monitoring page included a ‘View surveys’ button on the bottom:

“It took me ages to realise there was a ‘View surveys’ down there, so you could actually see what people had been say. . . you know, the results for the last few days”. (doctor from LUCS team, COPD and CHF)

The care provider did not offer an explanation for this problem, but I suspect that it was also related to the resolution of her screen and the fact that the website did not adjust at all to the size of the browser’s window, just adding scroll bars once decreased. A smaller resolution and the fact that the care provider was keeping a smaller window might have therefore hidden the button, which was placed at the bottom of the screen, from her sight.

3.8.4.1.4 Counter-intuitive Button Placement

The users’ difficulty in finding the ‘View surveys’ button described in the previous point could also be explained by an unclear positioning of this button, on the bottom of the screen, instead of at the top where it would have been more visible. Moreover, **other two users, one care provider and one CO, commented on the placement of the website’s ‘Back’ button.** The ‘Back’ button was placed on the top right hand side of the web page, and it was meant to be used for navigation through the website. One CO commented that she had found it hard to remember to use the ‘Back’ button of the system, and not that from the browser:

“(. . .) one of the major things, before going onto the telehealth system itself, erm, you always wanted to press ‘Back’ on the actual Internet Explorer but you’ve got your little ‘Back’ box in the [website], so that was one of the big things you always needed to remember to do.” (CO)

One care provider mentioned that she sometimes ‘mistakenly’ used the ‘Back’ button of the browser, instead of the website’s ‘Back’ button. As she explained, this would unfortunately lead to a functionality issue, as it would log her out of the system and thus delay her in her work (efficiency problem):

“Clearly the thing that really annoys me is that I keep shutting it down by mistake. (. . .) if I click ‘Back’ then I just go out of the system completely (. . .) I’m just used to, you know, going back to a previous screen directly from that, so . . . so it’s annoying that I’m not allowed to do that.” (respiratory physiotherapist from CRT team, COPD)

The users were not making a mistake, their use of the browser’s button being a

normal habit and, moreover, encouraged by the website's positioning of the 'Back' button. It is generally acknowledged, first of all, that a website should not have a 'Back' button as the browser already offers such a button to its users, which they are used to. Moreover, even if a 'Back' button is necessary within the website, it should at least be placed on the left hand side of the pages, as the users have the natural tendency to look for such a button there, while they have a tendency to look for a 'Forward' button on the right hand side of the screen.

3.8.4.2 Data Representation Problems

Good data representation is essential for care providers, as the way that information on the patients is organised, the details it provides, its colouring and textual/graphical presentation are very important for raising their awareness as to the patients' state and speeding up their decision-making (thus also influencing efficiency).

The following were the most important data representation problems identified by the users for the telehealth system:

3.8.4.2.1 Inappropriate Ordering or Structuring of Information in Tables

There were several cases in which users felt that the ordering of the information did not conform to their needs:

- **The patient list from the homepage was only ordered according to the time that the patients had sent in readings, while more than half of the care providers (7/13) preferred a default alphabetical ordering according to the patient names.** We have seen in subsection 3.8.3.1.2.1 on navigation that the absence of an alphabetical ordering of the list once sorted by teams was also causing inefficiencies and frustration to the care providers.

"If they were in alphabetical order, you're not spending your life doing that, looking for each patient, which is what we otherwise have to do." (doctor from LUCS team, COPD and CHF)

- **The different types of patient readings which were available to the users on the lines of a table on the Patient Monitoring page could appear in a random order there.** This was reported as a problem by the heart failure nurses, who would have preferred blood pressure and pulse readings to be presented in the order systolic-diastolic-pulse:

“The blood pressure is...for me, anyway, the wrong way round, because normally your diastolic is your lower number, the systolic is your higher number, so I find that difficult to read through when you’re going through (...) if you’re looking at a trend of blood pressures, that’s quite clumsy and difficult to read through when you’re...when you’re on the phone with somebody.” (nurse from Heart Failure Nurse Team, CHF)

“That’s all the wrong way around, it should be systolic, and then diastolic, and then pulse, and we all found that confusing.” (nurse from Heart Failure Nurse Team, CHF)

One of the nurses also added that her personal preference was to group the systolic and diastolic readings into the blood pressure type:

“It would actually be better if you had blood pressure and pulse as two separate... readings, as opposed to systolic, diastolic and pulse that day, but then that’s maybe my personal preference, but, you know, it would be quite good to go- that’s the pulse, that’s the blood pressure, that’s the pulse, that’s the blood pressure” (nurse from Heart Failure Nurse Team, CHF)

A similar comment on the order of reading types in the table was made by a care provider from a COPD team and by a CO, both referring to oxygen and pulse readings. Both of them, as well as another two care providers, also pointed out that readings were not always structured in terms of dates and times. This made it difficult for them to retrieve the latest readings for the patient, and to follow reading trends in case the patient repeated readings that day (efficiency problems):

“Cause the last... there was one of the... people I looked at today, has done readings four or five times... here... so there’s four oxygen and there’s four pulse, so... but if you look at the times, they’re... the times are bizarre times, ehm, so actually that one’s the earliest, but this one’s the most recent, so that’s... that to me would be the wrong way around.” (respiratory physiotherapist from CRT team, COPD)

“I think it was cause he had done two measurements today, or they come in today, you know he’d done them yesterday, I think there was two oxygen at the top, and then two pulse rates, so I had to look a wee bit to work out what time each one had been done, and mark out what was the most recent one.” (CO)

The care provider who commented both on structure and order suggested that

readings always be presented in pairs for the two types, and always in the inverse order of the time:

“Participant: If you could have them just together [the readings of the two types], cause that’s how we would record a reading, you record both together.

Me: Yeah. So always have them in pairs.

Participant: Yeah, yeah, absolutely.”(respiratory physiotherapist from CRT team, COPD)

“It would need to be... the most recent, top readings at the top, and, and there... and in pairs thereafter.” (respiratory physiotherapist from CRT team, COPD)

The CO suggested having different columns for the different reading types:

“ it might be easier if they were... separated out, so you just had one column for pulse rate, day, time, measurement, and one column for oxygen, maybe it’s better, a little bit easier.” (CO)

The care provider suggested instead having them on the same column, but as separated into groups of reading types which have been taken at the same time:

“I think the date would be better first, and then the systolic and diastolic and pulse, and maybe all on the same... column and a space.” (respiratory physiotherapist from CRT team, COPD)

- **On the Alerts page, there was no differentiation between the alerts triggered by the patient’s readings and survey replies that day and those from the days before.** This comment was made by one CO, who explained that she sometimes wasted time (efficiency problem) in trying to address old alerts. She suggested having only today’s alerts open and an option to open those from previous days:

“(...) there’s times where you go in and work the alert assuming that it is today’s date, but you don’t realise that it was actually yesterday’s, and you don’t need to work it (...) So if you were able to just take them away, so if it’s just today’s alerts that came up, but there’s an option to look up previous alerts” (CO)

3.8.4.2.2 Inappropriate Highlighting of Information

Care providers commented about the need for highlighting for:

- **Patients who needed to be checked from the homepage by them:**

“If you go back to him [the patient] on the sheet, it’s still up as active, and the ones who are... ill today, or who needed a call as well today, like our man X there, it’s also just ‘active’, so there’s nothing (...) I can’t look at this sheet and think ‘that’s the one I need to do, that’s the one I need to do’”. (doctor from LUCS team, COPD and CHF)

Patients were highlighted on the homepage by using a red or yellow square, but this highlighting was only meant for NHS 24 COs, and it would disappear once the COs closed the alerts for the patients. As care providers sometimes did their telehealth monitoring while COs were still working on the system, they would also see these notifications for patients on the homepage, which were not meant for them. Two care providers told me that this was confusing for them, as they were not sure whether they should act on the alerting patients:

“If I don’t need to know about them, I don’t want them to be there. Cause I feel that I’m missing something and... should I be acting upon this? They’ve got two red marks next to this lady, now what does that mean?” (respiratory physiotherapist from CRT team, COPD)

During interviews, other care providers were also unsure about the purpose of these notifications.

Almost half of the care providers (6/13, one of whom as asked by me) suggested being alerted on the homepage as to the patients whom they need to deal with for the day, even if they also received an email from the COs for this purpose. Two participants explained that this would help them be more aware of the patients that they need to check:

“If it actually showed you... which patient had an alert for that day, you know, so if they came up red... (...) I think it would make it easier for me just for thinking ‘right, you know, have I dealt with everything for the day?’, and I could just look at it on one screen.” (respiratory physiotherapists from CRT team, COPD)

While the majority of these care providers saw this as a separate type of colour highlighting which would only appear for them, as part of their own interface, one participant explained that their team had already asked NHS 24 not to close off the alerts such that they could also see the notifications from the homepage and use them:

“And we have asked them before, can you not sign it off, so when we go in... cause

all that means is that the alerts... you know, that they've looked at the alert, it doesn't say whether the clinicians looked at the alert. (...) It would be good if, you know, the alerts stayed there so when we went in... you could see straight at a glance... and then you could just click on that, and get to where you want to be..." (nurse from Heart Failure Nurse Team, CHF)

While such a solution was useful for care providers, it was not accepted by the COs, as closing the alerts notified them when they had finished monitoring each patient.

Other care providers suggested being notified about alerting patients by having the website and email integrated. Their suggestions have already been presented in subsection 3.8.3.2.2.4.

- **The answers given by patients to the health-related surveys which were presented on the Surveys page** and which generated a total survey score for the users. Some of the care providers who commented on the representation of the patient surveys' questions and answers (3/11) explained that **it was difficult to understand what symptoms had triggered the score**, as it was a matter of reading through the answers, which did not stand out on the survey:

"What happens here it will say symptom score 3, but you have to read through it to find the one that's... low." (nurse from Heart Failure Nurse Team, CHF)

Two of them suggested having the alerting symptoms as highlighted on the survey:

"He's said 'yes' there, if somehow that was in bold or different colour or something would be helpful, I think." (nurse from Heart Failure Nurse Team, CHF)

Another care provider described to me her difficulties in comparing survey answers between consecutive days, which was due to the extra navigation and scrolling required for moving between the days to see the whole information (as presented in subsections 3.8.3.1.2.4 and 3.8.4.1.1). She suggested that the surveys also include an indication of the questions which have changed since the day before, so that presentation could improve her navigation:

"So, in a perfect system, it would maybe be... on today's survey, it would be saying 'question 1 is different from yesterday', 'question 2 is the same as yesterday', 'question 3 is the same', 'question 4 is different'." (nurse from West Lothian Intensive Case Management Team, COPD)

Another care provider had a similar suggestion- that of having a sign next to the questions indicating the ones which have changed. When proposed by me, a third care provider also agreed that such guidance would be useful.

- **The alerts triggered for the patient, which were presented in the table in the Alerts page, and which had priority bands coloured in red or amber attached to them.** The colour of the bands was computed according to the number and criticality of the different alerts. **Although a scaling of the colouring was used for the two colours, this did not differ between bands of the same colour and probably had no real purpose.**

Two care providers commented on the way that the coloured bands were represented and what they meant, considering that they did not clearly communicate any information:

“When I saw these alerts and little sort of red and yellow things I, I just find it really... I don’t know. It’s just being with all this sort of... is it information in code really?” (respiratory physiotherapist from CRT team, COPD)

Another care provider expressed her confusion as to how the colour was attributed to the alerts, as this was not made clear anywhere:

“Ehm, what I do not understand is the priority, how the priority works, ehm, I guess red is because the score is higher, but there’s 3 and it’s red and then there’s 5 and that’s still red, I don’t understand priority, I don’t... get how that works.” (nurse from West Lothian Intensive Case Management Team, COPD)

Yet another care provider mentioned still feeling concerned because there were red and yellow alerts left in a patient’s table from previous days.

“I think after worrying that I’m actually... gonna miss, you know, I will check something but I will miss closing it down correctly cause I’ve not clicked on the right thing. Because, for instance, that’s still red, so does that mean somebody has not... (..) if something’s still red does that mean that somebody hasn’t checked what they should have done or... they haven’t shut that down adequately?” (respiratory physiotherapist from CRT team, COPD)

The care provider was not sure whether the colour of the alerts would change once a decision was made on them, which was not the case. The issue here was that the representation of the alerts did not provide enough feedback that an alert had been dealt with, at least for this participant. The only feedback it offered

was that the status changed, while maybe a bit more visual differentiation of the alerts which had been dealt with from the remaining ones would have been useful. As the problem was only reported to me by one participant, more research was necessary to understand whether this was the case for everybody.

The care providers did not have suggestions regarding the highlighting of alerts.

3.8.4.2.3 Inappropriate Colouring for Distinguishing Different Information

The measurement types from the Patient Monitoring page were each presented in a different font colour. A few care providers (3/13) and one CO considered that this was not ideal. One of them, a heart failure nurse, who also suggested having systolic and diastolic values paired together, mentioned that it would be useful if they also shared the same colour, and pulse a different one:

“(...) systolic diastolic in the same colour, finally have pulse in a different colour”
(nurse from Heart Failure Nurse Team, CHF)

Another care provider and one CO considered that the red colour was not a good choice as it could alert them, making them think that the reading was outwith threshold:

“Ehm, red and green? I wonder if red, at first ‘owww, what’s that’? ‘Something’s wrong’, have a look and that’s not wrong, and then I realised it was red and green... so, I don’t know.” (respiratory physiotherapist from CRT team, COPD)

3.8.4.3 Problems with Button Names

Buttons, as well as links, constitute the way to navigate through a website, and their intuitiveness is therefore essential for making it clear to the user where the information and functionality that they need lies, and how they could get to it quicker. Button names thus also become an important design aspect for ensuring efficiency.

Half of the care providers who participated to my study and commented on buttons (6/12), and two COs (2/13) considered that the names of the buttons from the homepage were not clear. Most of the care providers (5/6) mentioned needing to remind themselves from time to time what each button was leading to:

“Me: Are the buttons for each patient, are they immediately obvious to you which one leads to which information?”

Participant: These buttons? No, no. It’s all down to... muscle memory, I think... yeah, that’s completely what it’s about. If I go away for holiday for 2 weeks, yeah, I could get

confused again, I would be... sort of, jumping between Patient Monitoring and Alerts, and Programme Details to... remember, so, no, it's, no it's not, but it's just experience." (nurse from West Lothian Intensive Case Management Team, COPD)

Another 2 care providers also mentioned having learned about the buttons in time.

One CO also mentioned having had problems with them when she had first started using the system:

"Participant: I didn't know what was behind the... cause that's 'Monitor' and 'Patient Monitoring'.

Me: So the names aren't too... too intuitive there.

Participant: No. And then 'Programme details', it kind of didn't really tell you anything apart from 'Alerts', it's quite (...), but the rest didn't kind of tell you... what was in there." (CO)

The following issues were mentioned in relation to button names:

- **Not knowing where to find patient alert thresholds.** Two care providers mentioned expecting the thresholds to be available by clicking on the 'Patient Monitoring' button, and not the 'Programme Details' one:

"(...) just showing you there about... thresholds is under 'Programme Details', where... initially I felt that, you know, my brain was telling me it would be under 'Patient Monitoring'." (nurse from the EMACS team, COPD)

Please note that 4 other care providers also mentioned not knowing where to retrieve patient thresholds, even if not stating that this was due to the lack of clarity of the button names:

"I... used to be able to check it on the system, but I've now forgotten how to check it on the system, the threshold." (respiratory physiotherapist from CRT team, COPD)

- **Getting confused between the Patient Monitoring and Programme Details pages.** Maybe in relation to the comments from above, this was mentioned by two care providers:

"In some ways, you know, 6 months later I still have to think 'what was that again?', so still I don't think it's clear enough, cause you've got one called 'Patient Monitoring', then you've got one about... the programme, you know, as opposed to, you know, like- 'Patient...'" (nurse from Heart Failure Nurse Team, CHF)

One of the care providers was basing her comments on observations of her colleagues as well.

- **Not knowing where to suspend a patient (i.e. consider her as temporarily inactive if she was in hospital, travelling, etc.).** Five care providers mentioned that due to the way that the information was structured on the pages of the website, they were not sure how to suspend patients from the system, and one of them specifically blamed the poor labelling of the buttons:

“Participant: We usually have to go and look for suspend, to find where they’ve put it.

Me: Aha. So you find this hard to remember.

Participant: Yeah. . .

Me: Is it because of the names of the buttons which are. . . ?

Participant: Well, there’s so much in the buttons. . . ” (nurse from Heart Failure Nurse Team, CHF)

One care provider suggested that having explanations show up when hovering the mouse over buttons would help her remember what these buttons lead to:

“I would guess, if you could maybe just put your, your mouse. . . over the button, and for it just to give like a list of options that that button would contain, so without actually having to maybe click into it to discover- ‘Alerts’ shows you that. . . I mean I don’t have the lingo for what this page is called, but if. . . if I could just hover over that, there must be a. . . a computer term for that, and it just gives you the options, that could be very helpful, yeah. It would. . . potentially save time as well.” (nurse from West Lothian Intensive Case Management Team, COPD)

The care provider also made the suggestions of prioritising the buttons and restructuring the web pages.

Another care provider suggested that it would be useful to change the name of the ‘Programme details’ button:

“(. . .) if you could start with the ‘Programme Details’ being called ‘Patient Information’ . . . ” (nurse from Heart Failure Nurse Team, CHF)

3.8.5 Content Problems

Healthcare workers do not only need clear, well structured information which is placed precisely where they need it, but also that this information be strictly the information that they need for their work, as any additional information may cause confusion, inefficiency and frustration.

The main problem with the content of the telemonitoring website was that it only had one interface to be used by all of the teams, resulting in each team having more information than they needed available. Moreover, this interface sometimes also included some old or unnecessary data. The participants commented on this as posing them problems in the following areas:

- **The patient table on the homepage, which contained patients for all of the teams.** One care provider explained that it was sometimes tiring and off-putting for her to see other teams' patients on the homepage:

"As, you know, I'm a slightly old-fashioned nurse, it has always drummed into me 'you never know anything you don't need to know', so it's like 'aww', yeah, and seeing those names is a bit difficult sometimes." (nurse from the EMACS team, COPD)

The great majority of the care providers who suggested the alphabetical ordering of the patient list (5/7) also suggested that this list be team-specific. Moreover, 3 other care providers mentioned this solution, only one in response to my suggestion, which made this idea desirable by more than half of the care providers:

"If, when you logged you could just get our patients rather than all the other... you know, EMACS and all the other... teams who are using it as well..." (respiratory physiotherapist from CRT team, COPD)

Two of the care providers explained that this would become especially relevant when scaling up the service:

"It would help if it could be service specific (...) especially if they are looking at us having, you know, hundreds of people on here, then that would be really good to have it, yeah." (respiratory physiotherapist from CRT team, COPD)

This problem can also be associated with the problems of searching for a patient, given the problems with the default ordering and sorting options presented in other sections. Team-specific interfaces were a potential solution for eliminating the need for a sorting option by teams, and result only in a need to have the patients as

ordered alphabetically by name by default. Also, they would have allowed for the highlighting of patients whom the different teams needed to check with different purposes, individually by teams as well, and eliminated the problem with this highlighting only being meant for NHS 24 COs (from subsection 3.8.4.2.2).

- **The alerts from the Alerts page, which when needed to be closed down by COs, offered them much more options for actions than they needed.** These options were only useful for care providers, or had been left from previous versions of the system. This was pointed out by 3 participants to the first part of the study:

“The status I know we... don’t use all of them (...) we don’t use ‘awaiting NHS 24 supervision’. The care providers do use their ones, erm, but we don’t use the ‘Pending’ ones either or ‘Unable to contact’, because we just assign it to care provider” (NHS 24 team leader)

- **The alerts from the Alerts page containing much more details than care providers needed for their assessment.** One care provider, who was very experienced with the system, felt that these details were only useful for NHS 24 COs for following their protocol, and mentioned not normally using the Alerts page because of this (please note that her colleague which I interviewed used the page, but it is possible that other colleagues did not):

“But the Alerts page, like I say, is... it really is designed for... and it has been designed around NHS... NHS 24, and their outcomes.” (nurse from Heart Failure Nurse Team, CHF)

“None of us really use that Alerts page, I know... I think the respiratory team tend to close the alerts, but we don’t, because I just... it’s not, it’s not well thought out... programming, it’s... doesn’t bring for us any... doesn’t do anything for us.” (nurse from Heart Failure Nurse Team, CHF)

Another 2 care providers commented on just how detailed the alerts were, and one of them explained that this could be useful for less experienced clinicians, but not for her:

“It’s just trying to bring together all the clinical reasoning that you’re trying to do, eh, and it seems long... long-winded the way it says that the threshold, or the saturation rate of 90 is below the threshold of... it’s in a very long-winded way. (...) That’s probably because we’re... fairly experienced users of that, and

are fairly confident with the system. I think for new users or pat. . . people who, clinicians who don't have much experience that. . . ehm, they may find that useful." (respiratory physiotherapist from CRT team, COPD)

Other problems with content had to do with the existence of too much content, or the absence of needed content:

- **The patient taking her readings and replying to her survey questions at one time could trigger several alerts on the system for the same time, all of which were represented as separate in the table of alerts from the Alerts page. Moreover, if the patient repeated readings and/or replied again to surveys, this would lead to another set of alerts for the same day. Almost all of the care providers who commented on the representation of alerts (11/12) considered that this was not useful, and moreover, that it was problematic, because:**

- Most of the participants (8/11) explained that they only needed to make one decision for the alerts, which they were only making for one alert (as also confirmed by observations) and therefore it was not useful to have all of the different alerts as separate, or even to have multiple alerts:

"There are too many alerts there, you could generate, so four alerts on any one day but we're only ever acting upon. . . any one of them, if you see what I mean, I just find it's duplication." (respiratory physiotherapist from CRT team, COPD)

Related to this point, several care providers mentioned initially making a decision on all of the alerts for the day, due to the existence of multiple alerts:

"When we first came out, we were told we have to close every one off, so you are going in and clicking off every one that was red for a set day." (respiratory physiotherapist from CRT team, COPD)

The conclusion is therefore that the way that the alerts table had been designed did not correctly take into consideration how care providers would normally need to use it to make a decision on alerts, which had initially led the users into using it inefficiently.

- Several care providers (4/11) explained that having to look over all of the little 'bits' of information for a day and time was sometimes off-putting to

them:

“Sometimes having that diluted by 2 or 3 different alerts can be quite off-putting.” (respiratory physiotherapist from CRT team, COPD)

“(...) just seems to be a lot of sort of... an awful lot of little bits that I’ve got to take notice of.” (respiratory physiotherapist from CRT team, COPD)

As a conclusion to my observations, some of the care provider teams had more recently instructed the care providers to only enter a decision for one alert, and the other care providers whom I have observed had decided to work this way. Nevertheless, when asked by me, a few of them were not sure that they were acting correctly. This uncertainty was partly caused by the existence of multiple alerts for one day on the Alerts page, as explained by two care providers, who stated that that they would sometimes get confused as to what they needed to do about them:

“I kind of look at and I think- ‘what I am supposed to do?’, and then I have to double check with somebody and I’ll say ‘Do I have to close them all?’, ‘No, no, no, just do one, just do it once’ (laughing).” (respiratory physiotherapist from CRT team, COPD)

Several care providers (4/11) suggested that having only one alert containing information from several would be an improvement. When proposed by me, another two participants agreed.

“Participant: I think one, one alert per day, is the patient alerting or not, you know, to even, sort of, for it to be that sensitive would be... for me, anyway, more preferable, you know, I just, I want a system that I can go into, see who’s alerted, you know, I need one alert to make a decision on, cause we... I’m working at a fairly high sort of clinical level, that we’ve got a capacity to do that.

Me: So would that alert be a summary of the three, so just say survey score...

Participant: yeah, even a summary of the three with a red dot, summary of the three with a yellow dot or whatever it is the one they put on there.” (respiratory physiotherapist from CRT team, COPD)

The rest of the care providers, when suggested by me, agreed that a functionality of closing down all of the alerts automatically when closing one would also be useful.

As compared to care providers, COs were instructed to always close all of the open alerts for a day. Despite not referring explicitly to the content of the Alerts page, some COs (3/7) and one of their team leaders also found that this was time consuming, and not useful. They all suggested having the functionality of closing down the alerts for the whole day instead:

“I think if you’ve got three to close down, if you close down one, you could close the whole lot, just finalise it, cause you’re doing the same thing” (CO)

- COs would normally write threshold values in their outcome on the Excel spreadsheet. Some of them (3/7), who used to consult the patient’s Monitor page for this purpose, found that **the Monitor page did not always provide a weight threshold for CHF patients, while it did provide thresholds for other measurements and for COPD patients**. They felt that this was an inconsistency which did not allow them to double check the weight threshold or to enter it into their comments while using this page. Moreover, one CO who had discovered that thresholds were also available in the Programme Details page described the navigation problem that this was leading to (as already mentioned in subsection 3.8.3.1.2.4):

“In the Monitor, yeah, you’ve got your thresholds, but you don’t have it for your weight, and the only way that I’ve been able to find a way out would be by going into their Programme Details to find out exactly what their threshold for weight is. (...) And that’s the only place that you could find it.” (CO)

Even more, the participant cited below seemed not to remember that she could scroll right into the alerts regarding readings from the Alerts page to retrieve thresholds, which confirms the presentation problem from subsection 3.8.4.1.1:

“There was one [patient] today, he triggered on weight and it said it was weight outwith threshold, so I went in to see what his threshold was, but he didn’t actually have a threshold on there (...) so we couldn’t really comment.” (CO)

Chapter 4

The Methodology

4.1 Overview

In the previous chapters, I have motivated the investigation of ways for predicting efficiency in different deployments of a telehealthcare system. This chapter illustrates a methodology that I have developed in this sense, for being able to make predictions in potential- real or hypothetical- deployments.

4.2 Basic Description of the Methodology

The methodology is intended as a guide for modellers for reusing models which have proven to accurately predict certain usability characteristics for a telehealthcare system and work process involving it within a reference deployment site, for predicting the same usability characteristics for the same system and its likely work process in a potential, real or hypothetical, site, based on the differences between the sites only. We define a **real potential site** as one where the system is planned to be deployed, or the feasibility of deploying it is considered, and where the modeller could gather information about likely differences to the reference site in terms of workload, user or work process characteristics. A **hypothetical potential site** is one resulting from the modeller's 'what-if' exploration of changes to a site, in terms of the workload, user, work process characteristics, or even system design. The changes can be both external foreseen changes, the effects on the usability characteristics of which the modeller wants to check, as well as changes proposed by the modeller to improve the usability characteristics or mitigate risks.

In some of the papers that I have published during my PhD ([28, 29, 26]), I have

only described the methodology as a guide for reusing models to obtain predictions of usability characteristics for real potential sites. In the current description, I will also consider the case of hypothetical potential sites.

The methodology is generic, in that it can be used with any modelling approach, and for predicting any usability characteristic which is relevant given the potential risks to the deployment. In this thesis, I propose the use of a modelling approach (described in Chapter 6) for predicting efficiency as a characteristic of usability and measure of success of telehealthcare system deployments.

The following is a description of the main parts of the methodology with their steps. The first part (the next section) is fixed, while the following two are interchangeable depending on the purposes of the investigation. In particular, the second part can be used for evaluating the usability of the system, in terms of certain characteristics, in a real potential site. The third can be used for investigating whether the system would still be usable from the point of view of the usability characteristics, or what would be the changes in the usability characteristics and how they could be counterbalanced, should there be changes to the site, which we consider transform it into a hypothetical potential site.

4.3 The Reference Deployment Site

We propose to first analyse the reference deployment site to gather information such as the following:

1. **Task analysis:** what tasks users aim to perform, when and how frequently
2. **Flow models of user interaction with the system:** what interface elements users use for their tasks, what can go differently than expected and how users react in such cases
3. **Errors that users could make:** their types and possible consequences
4. **Auxiliary systems:** when/how often users use other computer systems, paper, spreadsheets, telephone, fax, etc. or ask for help from other people for performing their tasks
5. **Numerical data:** e.g. number of patients users normally monitor, probability distribution of potentially critical (out of range)/uncritical readings for each

patient, number of failed/successful attempts of contacting the patient, average time the user talks with the patient, length and frequency of system delays, frequency of patient data inaccuracies observed by the user, etc.

6. User perception of the system: what leads to what degree of frustration for users while using the system and when

Depending on each type of data and the system's development stage, this analysis could be carried out in conjunction with the user-centred design of the system, during evaluation through usability testing or even in operational use. Information could be gathered using a mixture of interviews, log files (if available), documentation about the real users of the system, the system and any of its instrumentation.

To be able to decide on an appropriate type of modelling approach, during the study of the reference deployment site it is very important to understand what is likely to differ in new deployment sites, or how the reference site is likely to change, as well as what the major perceived risks are. No matter the chosen type of model, it must be useful for making verifiable/falsifiable predictions about certain usability characteristics which are of interest for addressing the associated found risks, be able to receive as inputs the information identified for the reference site and be configurable such that it could be easily reused to model different system deployments.

We instantiate the model (appropriately parametrised) for the reference deployment site and run it to make predictions of usability characteristics for the telehealth-care system and work process for that site. We then *validate the model by checking its predictions against reality, by carrying out a type of usability evaluation in the reference site which is cheap and reliable.* We *improve the modelling approach and revalidate the model against real data repeatedly until we consider the model good enough* to capture the reference deployment site's usability characteristics. We end up with a validated model of the use of the system and an understanding of the reference deployment site's usability characteristics.

4.4 Real Potential Deployment Sites

When the system is to be deployed into a real potential site, we *investigate the differences between the users and environment from this site and those from the reference site in order to change the inputs to our model*. Inputs might require simple numerical changes (e.g. in the median number of patients handled by the user) or more important

ones such as changes in the process and/or system functionality used by the users for accomplishing their tasks. The analysis could be carried out before the system is deployed in the potential site, for example, by interviewing future users, by sending out questionnaires to them, by speaking to their managers, etc.

Using the same modelling technique, we ***build a new model using the changed inputs for the potential deployment site***, and ***run the model to obtain new predictions about the usability characteristics of the telehealthcare system and work process for this potential site***. We can then ***compare the predictions for the potential site with the validated predictions from the reference site to better understand in what way the difference between the two contexts has influenced the usability characteristics of the telehealthcare system and work process there***.

This analysis could be conducted during the planning stage for deploying the system in any new site. My model-based approach could inform whether the system should be deployed in that site or not (i.e. whether it would be usable from the point of view of the usability characteristics). Moreover, it could help decide, on the basis of risk analysis, which is the best site from multiple potential ones where it would be best deployed.

4.5 Hypothetical Potential Deployment Sites

Depending on the perceived risks in each site, and the predicted efficiency of the system and work process there, we can decide to also investigate what would be the effects on efficiency of likely ‘what-if’ changes to the site in terms of the workload, user, work process characteristics or system design. Being hypothetical, information about such changes would not be collected empirically (as for investigations in real potential sites), but chosen based on perceived risks to stress test how the site would cope, or proposed to counterbalance the negative effects on efficiency of other changes.

Once ***a list of changes is decided on***, we follow the same steps as for predicting the usability characteristics in real potential sites: we first ***use the same modelling technique to build a model for the hypothetical potential site*** (incorporating the changes to the reference/potential site that we are investigating), and ***run it to obtain new predictions about the usability characteristics for this hypothetical potential site***. These predictions are by themselves useful for understanding whether the system would be usable from the perspective of the usability characteristics, given these changes. We can then also ***compare them with the predicted usability characteristics from the original (unchanged) site, to understand how the usability characteristics would be***

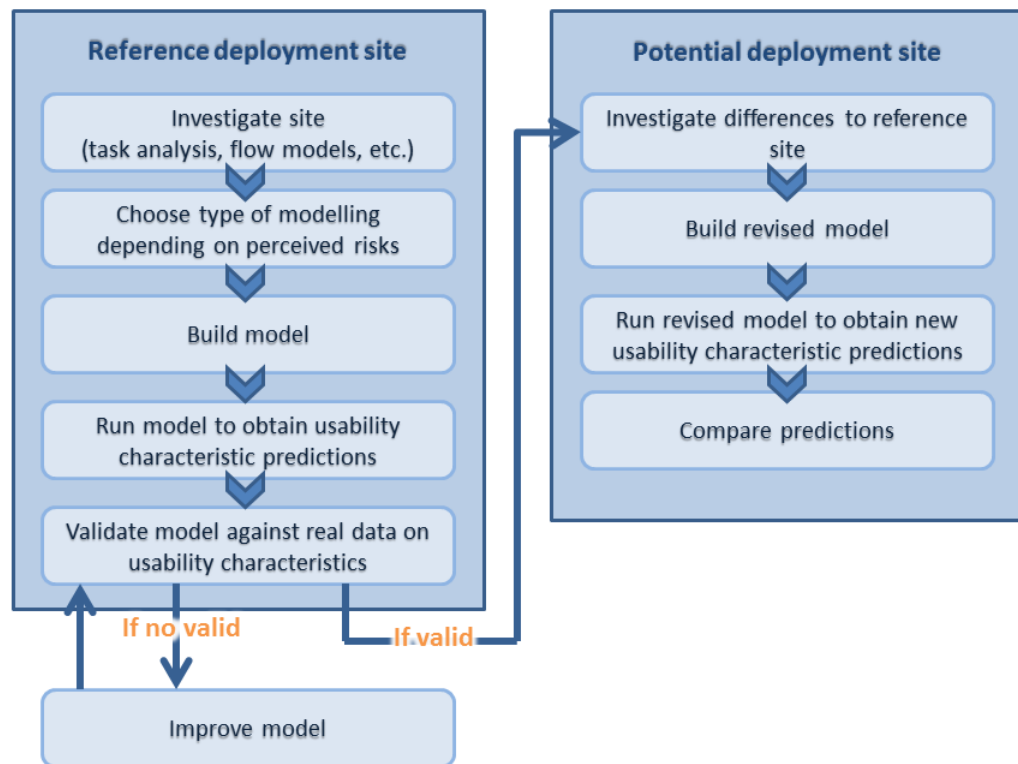


Figure 4.1: The Methodology

influenced by the changes.

Such an investigation should be best performed at regular intervals for existing sites, to help mitigate risks, and in the planning stages of deploying the system into real potential sites. It could have several purposes: understanding the limits of existing or potential sites, investigating what changes would be necessary for catering with the risks brought by foreseen negative changes, and investigating what changes would be necessary for improving efficiency in a real potential site where the system is planned to be deployed.

A graphical representation of the methodology, which abstracts over real and hypothetical deployment sites, is presented in Fig. 4.1.

4.6 Advantages of the Methodology

The methodology offers the following important advantages:

- It reduces the need for the costly usability testing of the system and work process

within each potential site, thus saving resources.

- It helps motivate investment in the real potential sites where it is predicted that the system is usable from the point of view of the usability characteristics.
- It helps identify the specific changes to the workload, user, work process characteristics or system design that are needed for making the system and work process usable from the point of view of the usability characteristics in a real potential site, in case it is otherwise found to be unusable.
- It helps understand the limits of current settings, in terms of workload, user, work process characteristics, and system design, between which usability is possible from the point of view of the usability characteristics, for current or real potential sites. Outside these limits lie the risks to the deployment.
- It helps mitigate risks and identify specific changes to the workload, user, work process characteristics or system design that are necessary for dealing with other foreseen negative changes.

4.7 Preconditions for the Optimal Use of the Methodology

For benefiting from the above advantages of the methodology, the following points should be taken into consideration:

- The appropriateness and quantity of the data collected for each type of input of the model influences the reliability of the obtained usability characteristic predictions. However, it may not always be feasible to collect the ideal amount of data of each type, and thus factors such as time, costs and the required result reliability need to be considered in deciding on the type and quantity of the data to collect.
- To make data collection possible, permission from the NHS to conduct a complex enough study methodology, involving all of the needed data collection methods, is necessary. Moreover, a good collaboration should be established with the organisations which will deploy, or already deploy, the telehealthcare system.
- As related to the above points, it is desirable to have access to large enough deployment sites, in order to be able to collect the needed data for the model

inputs for obtaining reliable enough usability characteristic predictions, given time and cost considerations. For the reference deployment site, there is a second reason for a large enough size: to allow for the validation of the model against enough real data on usability, and thus help develop the modelling approach until it is validated. The size of the deployment, and having enough data available, is therefore especially relevant for the reference site.

- The availability of realistic information about the differences between the users and environments of real potential deployment sites and the reference site is key for obtaining usability characteristic predictions which will indeed apply when the system is to be deployed within each of the real potential sites. The less realistic this information, the less accurate the results will be. The management and users of the real reference deployment sites should therefore be dedicated to helping in this investigation of differences.
- As related to the above point, it is best if the methodology is implemented for potential sites which have used telehealthcare systems in the past, so that management and users can use their experience to foresee how they would work differently with the system. While computer systems should be shaped according to user needs, and are thus influenced by users, professional systems such as telehealthcare systems also influence users and their work practices ([124, 167]).

The points regarding data collection from the above description are further discussed in the light of the work done for developing examples for this PhD, and recommendations are made for their implementation, in Chapter 10.

Chapter 5

An Initial Modelling Approach

5.1 Overview

Out of the types of usability problems which could be explored by means of modelling following the methodology presented in the previous chapter, I initially decided to focus on those which are related to mode confusion. *Mode confusion* is a user's misinterpretation of the system state, or incorrect expectation about its behaviour, during her interaction with the system. In a telehealthcare context, my motivation for choosing to look at this type of usability problems was that mode confusion can lead to the user being less efficient and making mistakes which can potentially lead to incorrect decisions putting patients at risk. An example of a non-critical mistake caused by mode confusion, taken from my study of the telehealth system used in Lothian, is the case of many users going through a sequence of web pages to save some key information that they had added about the patient on the first page, when in fact the system was already saving this information when they were accessing the second page. The design causes leading to the usability problem were the absence of a clear button to save the information, and/or of feedback about it being saved. As a result, users were wasting time by going through all of the pages, and were frustrated with the saving functionality. A user who knew how the saving functionality actually worked told me that she would expect the 'Back' button to save the information as well, which was in fact not the case. When the 'Back' button was pressed, there was no system feedback, and this could have led her to believe that the information was saved, when in fact it was not. This constitutes a more critical mistake caused by mode confusion.

I therefore built my initial modelling approach with the main purpose of it predicting mode confusion, and its design causes. Later, I realised that problems related to mode

confusion were not that frequent in the systems that I had investigated for my MSc ([25]), or in those that I was investigating during this PhD. Moreover, my study revealed just how important the issue of efficiency is for the users of telehealthcare, with respect to both the system and the work process surrounding it. This conclusion, as well as some disadvantages of the initial modelling approach, led me to build a second, improved, modelling approach, to be able to predict efficiency. This improved modelling approach is described in the next chapter. Despite the initial modelling approach being created for different purposes, it is still worth describing it, as its implementation led to lessons towards building a better approach for modelling telemonitoring work.

In this chapter I will motivate my initial choice of a modelling approach for predicting mode confusion, describe my implementation of this modelling approach, and highlight the disadvantages that I identified during this implementation and the general conclusions drawn.

5.2 Initial Choice of a Modelling Approach

My initial modelling approach was inspired by the PUM approach presented by Blandford et al. ([54, 46, 47]), which was briefly described in section 2.5.2.2.2 of Chapter 2. I chose PUM because of the following advantages highlighted by its authors which make it very relevant for the identification of mode confusion and its causes [47]:

- It focuses on the interaction between the user and a computer system as an object of study, unlike other modelling approaches from HCI and Software Engineering which focus on either the user or the computer system. Mode confusion is best identified by examining the interaction.
- It analyses the way in which users use their knowledge in the interaction. It is the lack of correct knowledge about the system which leads to mode confusion. The approach can also help predict goal achievement, reasons for human error and efficiency (which Blandford et al. define in terms of effort).
- As related to the above, it operates representations of the user and system in parallel, which allows for the check for mode confusion at every step of the interaction.
- By working at the rational, knowledge level (as defined by Newell [163, 164]), and not the cognitive level, PUM makes use of simpler modelling. Butterworth

et al. have shown that such simple, more abstract, modelling is sufficient for identifying potential issues with the design of a computer system [55, 47].

5.3 Description of the Initial Modelling Approach

5.3.1 Overview

I implemented the framework of human cognition and problem-solving from PUM, and adapted it to better suit the requirements of the user of a telehealthcare system. My framework simulates the goal-driven behaviour of an experienced user on a telehealthcare system and within her environment in parallel with that of the system (here called device), each essentially represented as a labelled transition system (LTS). This allows us to see where mode confusion, and other usability problems, may show up as the user being unable to achieve her goals. The implementation was made in Java.

In the following subsections I will provide a formal description of the notions and behaviour of my framework, and motivate the changes that I have made to the initial framework described by Blandford et al.

5.3.2 Notions

The framework models the system as consisting of two main parts: **the user** and a **passive device** (which only responds to user input). As my framework analyses a parallel view of two LTSs, one for the user and the other for the device, I will describe the two notions by using the classic definition for LTSs.

The device d is defined by means of a **set of device states**, a **set of named actions**, representing all the actions which are ever available for the user on the display, and the **transition relation** \rightarrow by which an action leads from one device state to another:

$$d = (DeviceState, Action, \rightarrow)$$

$$\rightarrow \subseteq Action \times DeviceState \times DeviceState$$

If $a \in Action$ and $ds1, ds2 \in DeviceState$, $(a, ds1, ds2) \in \rightarrow$ is written as $ds1 \xrightarrow{a} ds2$.

The notion of device components is important for understanding that of device states. A **device component** is an element of the device which may be observable by the user (e.g. the mouse, the webpage, a table), or not (e.g. information which is saved

in the system's database but not graphically presented to the user). At any one time, each device component can be in one of a set of states (e.g. the mouse can occupy a certain position within its available positions on the screen).

$$\text{E.g. } DeviceComponent1 = \{state1, state2, \dots\}$$

The set of device states can be represented as a subset of the Cartesian product of the sets of states that the device components can take. It is not necessarily equal to the Cartesian product, as some of the combinations of the device component states might not be possible.

$$DeviceState \subseteq DeviceComponent1 \times DeviceComponent2 \times \dots$$

At any moment the device is in a certain device state. One **device state** ds_x is represented as a **tuple of device component states**.

$$ds_x = (state1, state2, \dots), \text{ where } state1 \in DeviceComponent1, \\ state2 \in DeviceComponent2, \text{ etc.}$$

In each device state only certain actions are available.

The user u is defined within my framework by means of the **a set of user states**, a **set of named operations**, the **transition relation**, by abuse of notation also called \rightarrow , by which an operation changes the user state into another user state, and **an available time limit** $time$ (notion which was added by me to reflect the user's total available work hours):

$$u = (UserState, Operation, \rightarrow, time)$$

$$\rightarrow \subseteq Operation \times UserState \times UserState$$

If $op \in Operation$ and $us1, us2 \in UserState$, $(op, us1, us2) \in \rightarrow$ is written as

$$us1 \xrightarrow{op} us2.$$

In practice, my LTSs are deterministic.

A **user state** can be defined by means of a **belief state** and a **goal state**:

$$UserState = BeliefState \times GoalState$$

A **belief state** can be defined by means of **lower granularity belief states corresponding to the device (the device belief state)**, **work environment (the external belief state)** and **the user herself (the internal belief state)**.

$$BeliefState = DeviceBeliefState \times ExternalBeliefState \times InternalBeliefState$$

At any moment, the user may not know precisely what state the device is in. However, she may be able to rule certain states out. A **device belief state is a set of device states, representing the set of device states that the user believes the device could currently be in.** Leaving aside for the moment the possibility that the user is mistaken (which will be discussed in the next subsection), if the user's device belief state is b and $b \subseteq DeviceState$, then the true state of the device is an element of the set b .

A common case is that the user will have exact knowledge of the states of some device components and no knowledge about the states of other device components; this gives a compact way to represent a device belief state.

Elements of $ExternalBeliefState$ and $InternalBeliefState$ are also sets, of external and internal states respectively. Since we will not need to consider dependencies between device belief states, external belief states, and internal belief states, we will identify the belief state (d, ex, in) with the set of triples of concrete states $d \times ex \times in$. We will also write statements like “the device state dd is in the belief state b ” where what we strictly mean is that $b = (d, ex, in)$ and dd is in d , where no confusion can arise.

Each **goal state** can be defined as a list of individual goals:

$$GoalState = Goal\ list$$

We need a list of goals because the user will need to set up a main goal and subgoals within her goal state during the run of the operational model which instantiates the framework (as I will explain below).

In the same manner in which I used the notation of **X list** to represent a list of elements of type X , in the following definition I introduce the additional notation **X optional** to represent a set of elements of type X of size 0 or 1.

In my implementation, a **goal** from the goal list can be defined as **the tuple consisting of a belief state $BeliefToAcquire$ and an optional operation (commitment)** which could satisfy the goal but is not possible at the moment. The commitment corresponds to an element (individual operation) of the whole set of operations available to the user:

$$Goal = BeliefToAcquire \times Operation\ optional$$

As for any belief state, **$BeliefToAcquire$** can be represented by means of **lower granularity device, external and internal belief states**. Its device belief state, which

is the most important for the functionality of the framework, represents the user's intention to bring the device within a subset of the set of all possible device states, namely, the set of device states that define this device belief state. The simulated user will consider the goal to be satisfied when her belief state becomes a subset of the *BeliefToAcquire*. That is, even though the user may still not know exactly what state the device is in, she knows that it is in one of the states that she was aiming for.

An **operation** represents something the user believes she could do, for some purpose (e.g., to satisfy a particular goal), under certain conditions. Each operation has associated the following types of information:

- **Purpose** *P*: represented as a belief state, helps decide which goal(s) the operation is appropriate for ([47]). Each operation must have a purpose.
- **Precondition** *Prec*: represented as a belief state, records what it is that the user needs to believe in order for an operation to be achievable. Each operation must have a precondition.
- **Filter** *Filt*: represented as a belief state, records what it is that the user needs to believe in order for the operation to ever be possible. It is different from the precondition as a user can try to achieve a precondition (as will be explained in the following subsection), in which case the operation will become possible once the precondition is achieved, while a filter not being true makes the operation completely impossible.
- **Tracking** *Tr*: a rule on changing the current belief state (thus also user state) to reflect assumptions about the effect of an intended device action (term which is explained below) or of an omitted (as explained below) physical/cognitive action.

$$Tr : Operation \times UserState \rightarrow UserState$$

$$\forall (bs, gs) \in UserState, \forall op \in Operation, \exists bs' \in BeliefState \text{ so that}$$

$$Tr(op, (bs, gs)) = (bs', gs)$$

- **Intended device action** *ida*: what device action (if any) the user wants the operation to trigger. In contrast to Blandford et al. ([47]), an intended device action does not necessarily have a real correspondent in terms of a device action (if the user expects an action to exist on the device when it does not) and, even if it does have a device action correspondent, this correspondent is not necessarily

available on the device at that time (if the user expects that an action should be available to her when it is not). I believe that this consideration is important as it can help highlight the user's unsupported intentions and unsatisfied expectations, thus revealing usability problems. Together with this change, I decided to also consider operations with no associated intended device action but of importance to us as they represent important steps in the user's work and take up an important part of a medical user's time: cognitive operations (e.g. assessing a reading or a history of readings) and physical operations (e.g. writing a comment). These operations would logically normally have associated cognitive or physical actions, but I decided to omit them, as actions which are not related to the device are not of importance for my analysis. The intended device action is therefore an optional piece of information, represented as a set of size 0 or 1.

- **Visible effect** Ve : notion taken from older work by Blandford et al. ([54, 46]), represents a rule on changing the current belief state (thus also user state) to reflect the visible effect of an intended device (and not physical or cognitive) action, if any.
- **Time** t : added by me so that the approach could also provide some indication of efficiency. It represents that maximum time that the user spends in performing the operation, on the device or in her environment, and should be collected empirically.

For any $op \in Op$, there is $Info(op) = [P, Prec, Filt, Tr, ida, Ve, t]$, where we define:

$ida : IntendedDeviceAction$, where $IntendedDeviceAction = Action optional$

$Ve : Op \times UserState \rightarrow UserState$

$t : float$

Each operation transition \rightarrow can in fact be represented as the composition of two transitions: the tracking and the visible effect, as explained in the following description of the framework's behaviour.

5.3.3 Behaviour

My framework is a discrete time model, with both the user and the device state being changed over time by transitions. Each simulation step takes place once an operation

is performed by the user. More exactly, given the duration of an operation (the time component of an operation's information) t_{op} and the time of the simulation step before performing it t_i , a new simulation step takes place at time $t_{i+1} = t_i + t_{op}$.

The behaviour of the framework can be summarised as occurring in cycles, where in each cycle the user attempts to achieve her **current goal**, the last goal in her goal state's list of goals. The purpose of the behaviour is for the user to achieve her initial goal (the first goal in the list), which is known in the work by Blandford et al. as the **principle of goal-driven behaviour** ([47]). The user starts in each cycle by looking for **rational operations**, i.e. those the purpose of which leads to the achievement of the current goal and whose filter is satisfied. An operation's purpose leads to the achievement of the current goal if it is a subset of the goal's BeliefToAcquire, i.e. if the operation could bring the user's belief state within the set of states required by BeliefToAcquire for achieving the goal. An operation's filter is considered as satisfied if it is a superset of the user state's belief state, i.e. her belief state is in one of the set of all possible states that the filter is requiring. Although Blandford et al. suggest that an operation could partially contribute to a goal ([47]), I found this notion difficult to represent, as how much an operation contributes to a goal could depend on user preference or intuition. Blandford et al. also added here that an operation is rational if it does not hinder the satisfaction of any subgoals that the user might have. In my implementation, this notion was intended as future work. There may be several operations which are rational, which is why the user would have to choose the one she would attempt to accomplish for satisfying the goal. Butterworth and Blandford propose several heuristics for doing this in [54], the exploration of which I also intended as future work. In my current implementation, the analyst (the user of the model) is asked to choose the operation she prefers. This allows the exploration of different user behaviours in different runs of the framework.

Once a rational operation is found, the user (the simulated one) checks if the operation is **possible**. This involves checking whether its precondition is also satisfied, i.e. its belief state is a superset of the user state's belief state. If the operation is not possible (although it was rational), the user adds it as a commitment to the current goal and the operation's components of the precondition as subgoals to her list of goals (which is known in the work of Blandford et al. as the **principle of commitment** ([47])). This initiates a new cycle where the user attempts to achieve the first subgoal, her new current goal.

If the operation is possible (and it was also rational), the user attempts to perform it (known as the **principle of immediate performance** in the work by Blandford et

al. ([47]), which involves transitions in both the user and the device LTSs in my implementation. If the operation is to occur on the device, the framework first checks if its **intended device action** *ida* is **possible** (definition added by me): if it has a correspondent in terms of a real device action $a \in Action$ and if, given the current device state and this found action, there exists a transition in the device. The fact that there might not be any corresponding action to the intended device action can at this point warn the analyst about the user expecting an inexistent functionality on the device. If there nevertheless exists a corresponding device action, the fact that there might not be any possible device transition given the current device state and this action can warn the analyst that the user is expecting a functionality which is not available at the given time. If any of the warnings from above are given to the analyst, the simulation time will not change as the operation is not possible on the device, at the given time or ever. If no warnings are given, the simulation time is updated with the duration of the operation (taken from the operation's time information) to indicate the new simulation step. If the new time of the simulation exceeds the time element from the user's definition, the analyst is warned about the user not being able to achieve her goal in the given timeframe. Otherwise, the found device transition fires to get the device into a new device state. For the user LTS, once an operation is performed, the Tracking rule *Tr* included in the information about the operation leads to a transition to a new user state. By checking whether the new device state is a subset of the new user's state's belief state (**consistency user state-device state**), we can detect any inconsistencies between what the user expected and what actually happened on the device. Once the Tracking rule has changed the user state, the Visible effect rule *Ve* included in the information about the operation leads to a second transition to another user state. This second user state can also be checked for consistency to the new device state to detect any user misinterpretations of what she observed from the new device state. The differences between the application of the Tracking and Visible effect rules can also be checked to indicate any inconsistencies between what the user was expecting from the device and what she has observed from the device (the user might correct her state by observing something which is different on the device than she was expecting or not). The performance of transitions on the user LTS conforms to the **principle of belief updating** of Blandford et al. ([47]), with the addition that in my framework I reconsider the visible effect rule from the authors' older work ([54, 46]). This principle does not have a correspondent in terms of device transitions for Blandford et al. as the authors did not model a user's work with computer systems. Moreover, all the consistency

checks were added by me as detectors of usability issues.

Once the operation has updated the user and device LTSs and all the conformance checks indicating any usability problems have been made, the user checks whether she believes that the current **goal was satisfied** (according to the **principle of goal-driven behaviour** ([47]), i.e. whether the belief state belonging to her user state is a subset of the current goal's BeliefToAcquire. If the goal was indeed satisfied, the user removes it from her state's list of goals. She then initiates a new cycle with the newly reached current goal (the goal from her user state's list of goals which was previous to the goal just removed).

Once a new cycle is reached where the current goal has a committed operation, the user checks whether this committed **operation is possible, rational** (as per the explanations above) and also **motivated** and, if so, performs it as above. An operation is motivated if its purpose was not already achieved, i.e. it is not a superset of the user state's belief state. Once a committed operation is performed, or the user finds that it is no longer motivated, the current goal's commitment is nullified.

The processing described above continues until either the user has managed to achieve her initial goal, which is indicated by the goal list becoming empty, or she has got stuck by not finding any rational operations to achieve a current goal, or a usability error or warning was thrown by the framework for consideration by the analyst.

In addition to the checks for usability problems described above, my framework also checks whether the user has to use remembered information in interacting with the device. If so, it calculates for how long, and for how many actions on the device, she has to remember this information. This is relevant because, especially if interruptions are possible, the user might forget the information, leading her to making mistakes. Thus, the user needing to remember things may indicate a usability problem. More exactly, the user state's internal belief state may record that the user remembers a fact; I call this mental information. Each time the Tracking rule causes a transition in the user LTS that moves it to a state the belief state of which does include mental information, the framework saves the current simulation time and the current device state, and initialises a counter over the number of device actions with 0. This counter is increased each time a device action is performed, until the first time the user's operation is one that actually requires the remembered information- that is, the information is mentioned in its

has changed in important ways (e.g. the user is on a different page in the system) since this mental information was saved. If so, a warning is triggered to the analyst about the

user needing to use remembered information in a different device state. My framework reports the two device states, the difference between the current simulation time and the time when the information was remembered, and the number of actions that intervened. The analyst can judge whether this indicates a usability problem.

5.3.4 Types of Problems Detected by the Framework

My implementation is an extension of the framework from [47] in that, as well as determining whether the user's main goal and subgoals can be achieved, it permits the automatic detection of usability problems. To sum up the description from above and provide further details, the following are all the types of usability problems that my framework can detect:

1. Problems which the framework alone can detect (without being obvious to the modeller):

- **Mode confusion problems, the primary aim of the modelling approach:**

The different cases, as represented graphically in Fig. 5.1, are:

- The device behaving differently than the user expected and the user noticing the difference and correcting her knowledge of the device state (inconsistency Tracking Tr - Visible effect Ve).
- The device behaving differently than the user expected, but the user not noticing the difference due to misinterpretation (consistency Tr - Ve , inconsistency UserState Us - DeviceState Ds after each), which could lead to her making mistakes.
- The user making new misinterpretations (inconsistency Us - Ds after Ve), which could lead to her making mistakes.
- The user attempting an unavailable device action (intended device action ida corresponds to unavailable device action $a \in Action$).
- The user getting stuck not knowing what to do next due to unawareness of the real device state (elements from the device state not in the user state's belief state, e.g. unclear button, and no operations available for achieving the goal).

- **The user exceeding her available work timeframe without managing to achieve her goal:** whenever an operation is performed, the value of its time parameter is summed up and compared against the user's available

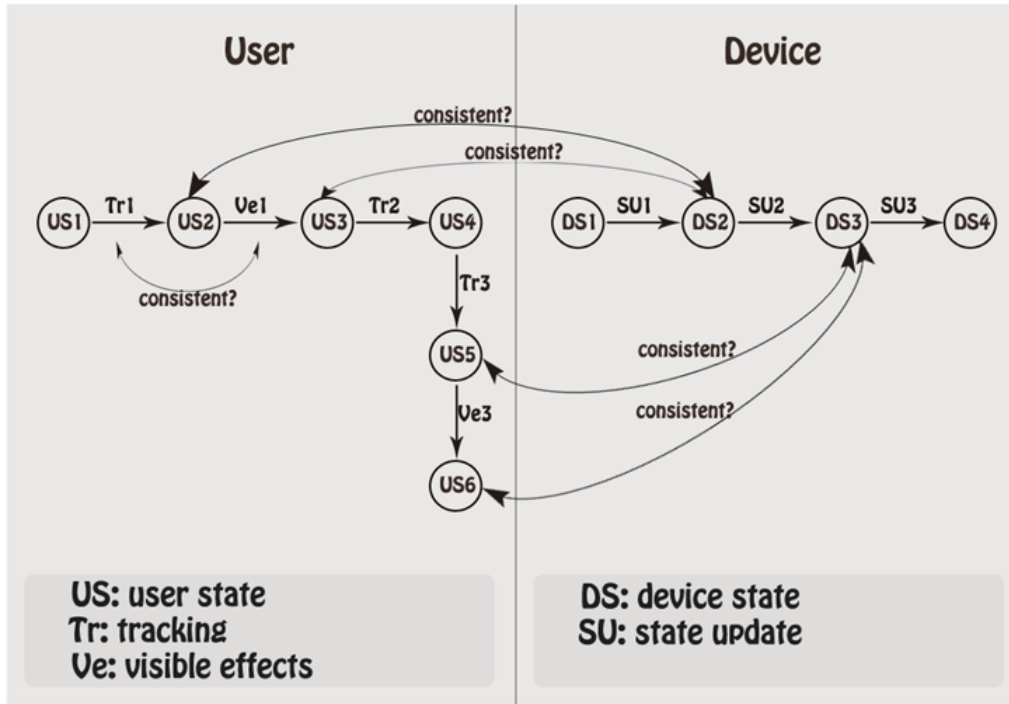


Figure 5.1: Mode confusion problems as detected by the controller

time limit, such that the analyst is warned if the user's available time limit was exceeded. This is useful for cases where repeated operations make it hard for the analyst to manually sum up the time.

- **User memory load:** it can warn the analyst about steps in using the device which require the user's use of her memory, which could lead to errors.

2. Unsupported user work, about which the modeller already knows before being detected by the framework:

- The user attempting to perform unsupported functionality (the intended device action ida does not have a device action correspondent $a \in Action$).
- The user using additional sources of information for attaining her goal (EHR, paper), which is apparent since the model is built.

We call the element continuously analysing the running of the user and device LTSs in parallel to detect any problems a 'controller'. The controller can be seen as sitting on top of the user and device parts of the framework.

5.4 Disadvantages of the Initial Modelling Approach

I implemented and instantiated the cognitive framework on a real case study of a telehealthcare system in use for monitoring diabetes as part of a Telescot ([21]) medical trial. The parameters that I provided for the framework were not derived from real-world data, but chosen by me based on experience from previous work [25] to be used for deriving an example model which can showcase the types of problems that the approach can detect. During the implementation and the running of the framework, I found the following disadvantages of the approach:

- The inputs required for populating the constructs for the user and device models need a strict format and are difficult to build.
- The user's belief state only holds generalised information (e.g. perceived patient), not distinguishing different instances (which can only be 'simulated' by means of operation parameters) or their characteristics/attributes (e.g. the measurement is red).
- The cognitive framework is simplistic and does not distinguish between different types of memory constructs (LTM, STM), perception or motion.
- Due to the simplistic cognitive framework, the approach does not determine the cognitive causes which lead to users having problems on the interface, which would be useful for knowing how to better correct existing usability problems.
- Like for other cognitive approaches, the analyst can find some of the problems while building the inputs for the framework, without needing to run it.

I decided to develop a more cognitively correct modelling approach which would consider the first three issues from above, and focus on the prediction of efficiency instead of mode confusion. In particular, I aimed to make the following improvements:

- Represent inputs by using a clear notation
- Consider perception as the means for populating user knowledge about the world
- Represent operation parameters (which will become percepts) in a way which is closer to reality. As a real user would distinguish instances of generic elements of the environment (e.g patients) which have certain types of characteristics with certain values (e.g. highlighted- red, name- 'John', age- 60, etc.), my framework

should also distinguish between general percepts and their instances and allow percepts to have different types of attributes with values.

- Consider short and long term memory
- Generally, analyse more closely the cognitive processes which take place for a user when she tries to achieve her goals. This requires a better understanding of the structure and functionality of the brain and its constructs and the consideration of these constructs within the framework.

Chapter 6

An Improved Modelling Approach

6.1 Overview

As motivated by the disadvantages found with my initial modelling approach, I decided to build a more cognitively correct and complete model for the user, by using as inspiration cognitive architectures. In this chapter I will describe the reasoning behind choosing a cognitive architecture as the basis for an improved model of the user. I will also provide an overview of the structure and functionality of my modelling approach. The readers who are interested in a more detailed description are referred to Appendix E.

6.2 Inspiration from a Cognitive Architecture

Out of my review of the literature on cognitive architectures, I found that the most relevant for my purposes are Soar ([134, 143, 133]), ACT-R ([34, 35]) and ICARUS ([139, 67, 137, 140, 142]), which were all described in Chapter 2.

All of these approaches represent the processes of perception, motion and different memory constructs (LTM, STM, WM) for the user, which I decided to also represent for my user model in the improved modelling approach. Both Soar ([134, 143, 133]) and ACT-R ([34, 35]) are very useful for developing learning mechanisms. However, I decided that, as my purpose is to simulate the routine work of users who are already used with the main functionality of the system, a learning mechanism would not be a requirement. ACT-R can also be used for modelling a simulated user's mechanisms of remembrance/forgetfulness. However, due to the complexity of this module, I decided not to use it for the current work, but to propose it as inspiration for future research.

For the purpose of proposing examples of use of the modelling approach, which I will describe in Chapter 9, I have instead developed more simplistic, but effective, mechanisms for remembrance and forgetfulness for the user.

Overall, I found that ICARUS ([139, 67, 137, 140, 142]) offers the best advantages for the simulation of a user's use of a telehealthcare system and therefore I decided to use it as the basis for building the user model in my new modelling approach, and to extend it where necessary for my purposes.

6.3 Advantages of ICARUS for the Purposes of this Thesis

The following are the most important advantages for which I have decided to choose ICARUS ([139, 67, 137, 140, 142]) as the basis for building my user model in the new modelling approach:

- Its representation of user inputs which details general types, instances and instance attributes. This allows us to model the user perceiving (and inferring beliefs about), for example, a certain patient with a certain name and a certain age, eliminating the disadvantage about the BeliefState only holding generalised information from my initial modelling approach.
- Its representation of concepts in LTM which can be used to model medical and practical knowledge (e.g. if any reading provided by a patient exceeds certain limits, it is an exacerbation)
- Given the available concepts and the current percepts, its development of beliefs in STM, which can be used to model medical conclusions (e.g. if we have a current reading meeting the conditions of the concept, i.e. exceeding the limits, the user believes that that certain reading is an exacerbation)
- Its use of heuristics when simulated users need to choose between different skills to perform in the world. In my initial approach the analyst (the user of the model) was required to choose an operation to be performed by the simulated user. The use of heuristics as in ICARUS allows the representation of different types of users with different knowledge (heuristics) in choosing skills and the uninterrupted simulation of their behaviour on the system.

6.4 Description of the New Modelling Approach

6.4.1 The Structure of the Models

6.4.1.1 Overview

From the lessons learned, I implemented an improved modelling approach in Java. The approach keeps from previous work the general view of the user and that of the system (here also named device) as two LTSs, and the notion of a controller checking their interaction. The user model is a cognitive model especially inspired by ICARUS ([139, 67, 137, 140, 142]), from which I adopted the representation of the main cognitive structures and the basic skill execution functionality. I am still representing the device as a classic LTS with states and transitions, where states are made up of the states of all of the elements of the display (e.g. page, menus, pointer, buttons, etc.) and functionality, and transitions are the actions which can change them.

I have enhanced the notion of the user with more detailed cognitive constructs. More exactly, I have added the notion of **memory** and distinguish, like in ICARUS, between STM constructs and LTM constructs. **STM** includes information which is more dynamic, and which the simulated user uses for accomplishing her goals. It consists of a perceptual buffer, belief and intention memory. **LTM** includes information which is more general and stable. It consists of procedural information (how to do things) in the form of skills- the skill memory- and semantic information (general knowledge) in the form of concepts- the conceptual memory. As inspired by ICARUS, constructs in STM are instances of more general constructs from LTM. I kept goals as a separate memory construct as in ACT-R ([34, 35]). I am considering that the list of goals and beliefs, as in my initial modelling approach, but also the other components constituting the user's STM, with which the user works at any one time, are part of the user's state. An overview of the structure of my cognitive architecture is provided in Fig. 6.1.

In the following subsections I will briefly describe each of the notions used for the user and device models. For a more detailed description, please refer to Appendix E.

6.4.1.2 Notions for the User Model

The **perceptual buffer** is one of the simulated user's constructs from STM, which consists of a set of **percepts**. Percepts can represent either elements from the device that the user has just seen (perceived), like in ICARUS, or elements from the device that she remembers. Some percepts can refer to information with which the simulated

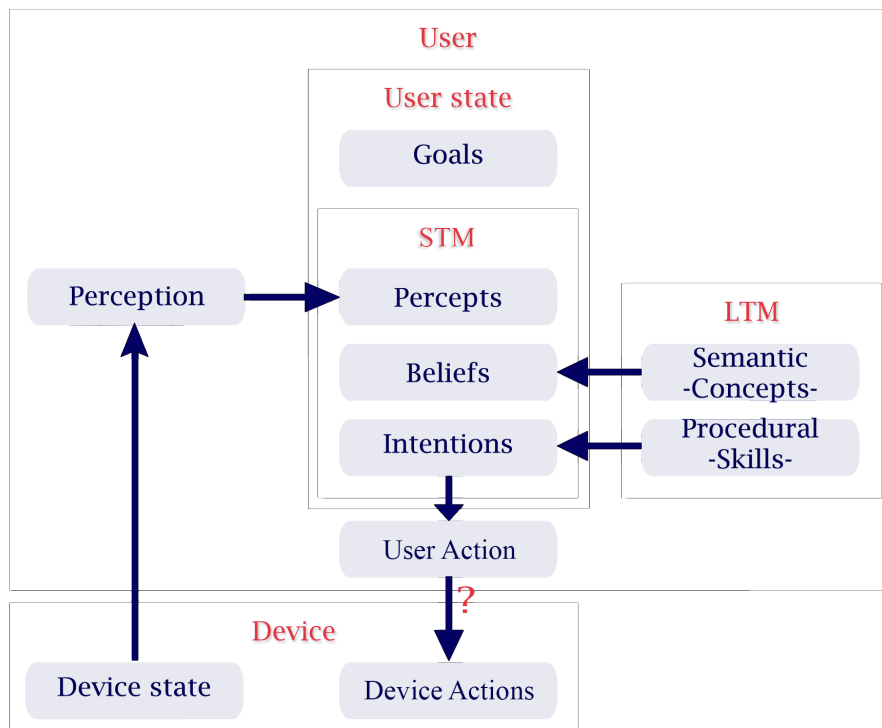


Figure 6.1: The structure of the new modelling approach

user works in her interaction with the system. For example, such a percept could refer to a patient (the percept's type) who has a name- "John", age- "51" and condition- "COPD" (the percept's attributes represented as name-value pairs). Other percepts can refer to the structure of the system. For example, we can have a percept of type "webpage", with a grey background (the percept's attribute). I have considered that a percept's attributes represent individual memory items that the simulated user can remember and forget. I will return to this point in the description of the mechanisms for remembrance/forgetfulness.

The **belief memory** is another STM construct. It stores a set of **beliefs**, representing inferences that the simulated user has made about the relationship between certain percepts, or the characteristics of one certain percept. For example, a belief could be that there is a relationship "placed_on" (the name of the belief) between a certain percept of type "patient" and one of type "webpage", or that a certain percept of type "patient" is suffering from COPD (e.g. this belief could be named "suffers_from_COPD"). Beliefs can refer to any percepts in my modelling approach: to those which have just been perceived from the system (as in ICARUS), but also to percepts which have been remembered by the simulated user (as an extension to Icarus). Beliefs must normally be instances of more general *concepts* from conceptual memory, which will be described

below. An exception is that, unlike in ICARUS, other beliefs can be directly added to the belief memory by the simulated user as a conclusion that *external skills* (skills performed in the static environment) have been achieved. I have not considered offering the modeller the option of also adding static beliefs, which do not need to be re-inferred, like in ICARUS. This functionality could be useful, for example to describe a clinician's prior knowledge about the patient's profile, and therefore I propose adding it as future work.

Percepts and beliefs are the most important notions for the user model, as they are also used by other notions.

For clarity, I will describe next the simulated user's LTM constructs, before returning to intention memory, the last STM construct. **Conceptual memory** is an LTM construct which consists, like in ICARUS, of a set of **concepts**, representing the generalised description of the inferences that the simulated user can make about the relationship between percepts, or the characteristics of a single percept. Concepts must define a set of *percepts*- general definitions of the format of the percepts on which they apply (as not specified by ICARUS) and of any percepts which must be present in the perceptual buffer such that they are applicable. For example, a "placed_on" concept would require a percept of type "patient" and one of type "webpage" for which it would apply, but it could also require a percept of type "table" for concluding about the placement of the patient on the webpage. Concepts may also define a set of *relations*- other concepts which must be applicable, by having belief instances in the belief memory, or not applicable if they are negated. For example, the "placed_on" concept from above could require that beliefs exist in the belief memory about the inclusion of the patient in the table, and of the table on the webpage. Lastly, as additional or not to relations, concepts may define arithmetic or logical *tests* over the attributes of the defined percepts which must hold such that they are applicable. For example, a "suffers_from_COPD" concept applied for a percept of type "patient" could require this percept to have an attribute named "condition" with the value equal to "COPD". Like in ICARUS, concepts which include a 'relations' component are considered as non-primitive, and form a hierarchy of concepts and subconcepts which depend on one another, while the ones which do not are considered as primitive and represent the leaves of this hierarchy.

The **skill memory** is another LTM construct which consists of a set of **skills**, subdivided into **device skills**¹ and **external skills**. **Device skills** represent the activities that

¹Please note that device skills belong to the model of the user, and not to that of the device. I have called them "device skills" to avoid a long name for them, e.g. "skills on the device"

the simulated user can perform on the device for attempting to achieve her goals, while **external skills** represent the activities that she can pursue in her static environment. This is different from ICARUS, which uses only a generalised notion for skills. In the following descriptions, I will always use the word “skills” to refer to both device and external skills, and only mention their type when necessary.

Device skills may require the presence of a set of *percepts* with a certain format within the perceptual buffer, in order to be applicable. For example, a device skill for accessing a patient’s Alerts page, with the name “access_alerts”, would require a percept of type “patient” on which to apply. Device skills should include a definition of a set of **conditions**- beliefs that the simulated user should have in her perceptual buffer/ not have if negated regarding the percepts, such that the device skill is applicable. For example, the “access_alerts” device skill could only be applicable if the patient needs monitoring (or has not already been monitored- negation) and is placed on the homepage, if only in such cases the simulated user would go to a patient’s Alerts page. Device skills should also include either a list of subskills, or an action. **Subskills** refer to other skills (device or external) which the simulated user must accomplish first in order to accomplish the device skill. For example, the “access_alerts” device skill from above could require the simulated user to first go to the patient’s Demographics page (“go_to_demographics” subskill), and from there to the Alerts page (“go_to_alerts” subskill) if the simulated user knows that the Alerts page is not directly accessible from the homepage. An **action** is the intended action that the simulated user would like to carry out on the device for achieving the device skill. For example, the “go_to_demographics” device skill could have an associated action of clicking on a certain button for the patient for accessing the Demographics page. Like in ICARUS, (in my case, device) skills which include a “conditions” component are considered as non-primitive, because they depend on other (device) skills for their achievement, and will form a hierarchy of skills and subskills during the skill execution behaviour which I will describe below. (Device) skills which include an “action” component are considered as primitive, and will form the leaves of this hierarchy. Device skills must always include a set of *effects*, representing beliefs that the simulated user is expecting to have in her belief memory, or not to have if negated, once the device skill is carried out. For example, as an effect to the “go_to_demographics_page” skill the simulated user would like to come to believe that the patient (i.e. the percept of this type) on which the device skill applies is on the Demographics page.

I have added to the general notion of skills from ICARUS the notion of **external**

skills in order to also be able to model external activities of the users in their static environment, but in a simplified way. In particular, while we are interested in thoroughly modelling a user's interaction with the system (here including by convention activities performed on other computer systems in relation to it), which may have unexpected effects, a user's activities in her environment (e.g. writing things on paper) would normally have the expected effects, as they are in her hands. I am therefore considering external skills as always directly achievable for the simulated user, while she will need to check the achievement of device skills. I will explain this in the section about the skill execution behaviour. Due to their direct achievement, external skills do not include "conditions" or "subskills" components. Their percepts act only as safety checks that the simulated user sees (i.e. perceives) certain percepts, as she would not need them for conditions or subskills. Also, I called the effects of external skills "true effects", to indicate that the simulated user will always believe in them, or not if negated, as an effect to the skill. External skills cannot have an associated action, as we are not interested in modelling the changes from a user's static environment. The simulated user will therefore add the beliefs from her "true effects" directly to her belief memory, as I will explain for the skill execution behaviour. As a skill in her environment is in the user's hands, as opposed to one on the system for which she must wait for system response, external skills for the simulated user also have an associated time. This time is defined by means of normal distributions (mean- standard deviation pairs) for normal, and interrupted, work, a chance of the simulated user to be interrupted, and a lower bound, representing the minimum value of the time. An example of an external skill is one in which the simulated user writes down a patient's name and condition on paper, named "write_down_patient_details", to which we attribute a certain time and the belief "details_written_down" as the true effect.

Both types of skills can also have attributed a set of *percepts to remember*, defining the format of percepts which the simulated user would like to remind herself of, and remember for a longer time, if she perceives them once the skill is achieved, so that she could use them for another skill. I will return to them in the description of the mechanisms for remembrance/forgetfulness.

The same way as beliefs are instances of concepts, intentions from the simulated user's **intention memory** STM construct are instances of skills from LTM. An **intention** represents the simulated user's planned next activity to pursue on the device (as an instance of a device skill) or in her static environment (as an instance of an external skill). To instantiate a skill, percept formats from its 'percepts' component are replaced by

specific percepts from the simulated user's perceptual buffer which match. For example, if the simulated user decides to pursue the device skill "access_alerts" described earlier for a certain patient which respects the format required by the skill's percept, she would create the intention with this specific percept. Unlike in ICARUS, I have not included in intentions for device skills the "conditions" component, as conditions are checked before instantiating the skill into an intention during skill execution, as I will explain for the skill execution behaviour. The same way as skills and subskills form a hierarchy, intentions and subintentions also do in the process of skill execution.

I have decided not to implement the functionality of problem solving of ICARUS in my approach, because the users from the telehealthcare context whom I have observed, even if beginners, were already used to their routine work which I modelled. Nevertheless, I have decided to use its notion of goals, and consider them as elements of the user's state, together with the STM. **Goals** represent, as in ICARUS, what the simulated user would like to come to believe as an effect of her skill execution behaviour. Goals are very similar to beliefs, and can be negated to show that the simulated user would not like to believe in something. Also, a goal in my approach can be made to apply to 'any' or 'all' percepts of a certain type, but only if it refers to one percept (unlike in ICARUS). I call such goals *repeatable goals*. A simulated user could have, for example, a goal of monitoring a certain patient, any or all patients, or of reaching a user state where the patient is no longer highlighted on the device.

6.4.1.3 Notions for the Device Model

The device is at any one time within a certain **device state**. A device state consists of a set of **device state components**, representing the different elements of the device's display and functionality, which may or may not be visible to the simulated user. Device state components look very similar to the simulated user's percepts. Also, they may or may not be 'on screen', i.e. rendered by the device, and 'visible', i.e. displayed within the simulated user's screen margins. Their similarity with percepts is due to the fact that the simulated user's percepts will be built from them during the process of perception, as I will explain in the next section.

The device also has a set of **device actions**, which define all of the existing actions that the simulated user could carry out on the device. Device actions are conceptually different from the actions that the simulated user can request from intentions instantiating device skills, which I call 'user actions'. This is because user actions might not have an existing device action correspondent, or a possible one for the given device

state. In order to be possible, device actions impose a set of **conditions** on the format of the device state in terms of its available device state components. For example, a device action for changing a webpage can require the presence in the device state of device state components describing a certain button, a certain current page, and a patient whose page is to be accessed. Device actions also have an associated **time**. In my implementation, due to difficulties in identifying user movement time out of the data collected for running the models, as I will explain in Chapter 8, I have taken the simplified view that the duration of device actions (i.e. their time) includes both the simulated user's movement time for doing the action on the device, and the response time for the device to change its state. It is defined similarly to the time of external skills, in terms of normal distributions (mean-standard deviation pairs) for normal and interrupted times, a chance of the simulated user to be interrupted, and a lower bound. I propose as future work the separate consideration of user movement times, either by using KLM ([61]), or as a separate component to device skills.

6.4.2 The Behaviour of the Models

6.4.2.1 Overview

The behaviour of the models can be summarised as follows: on a cognitive cycle, the simulated user perceives information from the device and adds it to her perceptual buffer in the form of *percepts*, process which I call **perception**. She then performs **conceptual inference**, by instantiating concepts into *beliefs* about these current percepts. Given her current goal or the stage that she is at for accomplishing it, and the current percepts, she chooses an appropriate skill and instantiates it into an *intention* in STM (**skill execution**). The skill- and intention- may be external, and thus be considered as always accomplishable, or on the device, and thus have an associated user action which is sent to the device with the help of the controller. Whenever it receives a user action, the device will use it to identify the equivalent device action, and use this device action to change the device state. The simulated user will perceive the new device state with the help of the controller and be able to check that her intention (and skill) were achieved. As separate, but interleaved within the behaviour for perception, the simulated user remembers a certain number of memory items for a certain time. She can also set herself to remember certain memory items for a longer duration, if she needs them for accomplishing her skills. This functionality forms part of her

remembrance²/forgetfulness mechanisms.

In the following subsections I will provide more details about the main processes underlying the behaviour. For an even more detailed description, please refer to Appendix E.

6.4.2.2 Perception

I have kept perception as very simplistic, as this was enough for the purposes of my examples. The simulated user always perceives all of the device state components which have been rendered on screen and are visible. Each of these device state components is converted into a percept. If the simulated user remembers having seen the same percept before (i.e. it is a remembered percept from her perceptual buffer), she reminds herself of it (more details in the section on remembrance/forgetfulness mechanisms). If not, she adds the percept as a new percept to the perceptual buffer.

The current perception functionality is separate within the implementation of the simulated user's behaviour, and could be easily replaced. As the current implementation does not deal with user eye focus, eye movement, attention, or assign a duration to perception, I propose as future work the collaboration with a psychologist for developing a more psychologically correct functionality.

6.4.2.3 Conceptual Inference

Conceptual inference is the process by which a simulated user instantiates generic concepts from her conceptual memory into beliefs to add to her belief memory, by matching percepts from her perceptual buffer to the concepts' percepts.

The process starts once the user has added new percepts to her perceptual buffer through perception. The simulated user takes each concept from her conceptual memory in turn and attempts to find percepts from her perceptual buffer which would match with its percepts (i.e. with their formats) to form what is called a 'binding'. Different bindings, representing different combinations of percepts taken together, could thus be generated. The simulated user tries the concept's relations (if the concept is non-primitive) and any tests with each binding in turn. Relations hold given a binding if their beliefs, as instantiated with the real percepts from the binding, exist, or do not exist if negated, in the simulated user's belief memory. Tests over percept attributes hold if their

²I decided to use the name "remembrance" instead of "memory" in this case to distinguish between the functionality of remembering information, and the two types of memory- STM and LTM, which represent components of the user model

arithmetic or logical relations are TRUE when these attributes are replaced with the attributes of the real percepts from the binding. If any of the concept's relations or tests do not hold, the simulated user tries out the next binding. If there are no more bindings, the concept is considered as not applicable, and the simulated user tries out the next concept. If all of the existing relations and tests hold, then the concept is applicable and the simulated user creates a belief which she adds to her belief memory, if such a belief does not already exist there. The creation of this belief may lead to other concepts depending on it through their relations to also become applicable and be instantiated into new beliefs and added to the belief memory. The process continues until no more new beliefs are generated.

Let us assume that we have a concept named "suffers_from_COPD" which requires a percept of type "patient" and with an attribute named "condition", and in its tests checks that the value of the attribute is "COPD". When the simulated user perceives from the device a real percept named, let us say, p1, with the required type and attribute, she will find that it is a binding for the concept, and then that it also respects the tests. She will therefore create a belief about p1 suffering from COPD, and add it to her belief memory.

6.4.2.4 Skill execution

Skill execution is the process by which the simulated user attempts to accomplish a skill, either as required by the modeller, or in an attempt to achieve her goals, as similar to ICARUS. In contrast to ICARUS, the latter option is not invoked from a mechanism of problem solving. This was not necessary for the modelled case study, where users were already used to the routine work that they were carrying out, and I never observed them getting stuck and needing to look for a solution. In my approach, the simulated user is expected to always know what to do to accomplish her goals, in the order in which they are given within a list (unlike the set notation used in ICARUS).

A high level view of the process of skill execution involves the following steps for the simulated user:

1. Pick the next unachieved goal from the list of goals, removing already achieved goals up to it. A positive goal is considered as achieved if it has an associated belief in belief memory, while a negative one, if transformed into positive (i.e. its negation is removed), should not have such an associated belief there.
2. Pick the next best skill to achieve the goal, based on a heuristic. The heuristic that

I implemented picks the skill which would achieve the highest number of other goals from the list of goals as well. A device/external skill achieves a goal if its effects/true effects contain a belief with the same name as the goal. If no skills are found to achieve the goal, an error message is returned and the run terminates.

3. Match the percepts of the skill with the percepts required by the goal and other percepts from the perceptual buffer to create potential bindings. If no potential binding is found, the simulated user goes back to point 2 to pick another skill.

For each potential binding:

4. Check that the skill was not already achieved. A device/external skill is considered as achieved if the positive beliefs from its effects/true effects have an equal equivalent in the simulated user's belief memory, and the negative ones, if transformed into positive (i.e. their negation is removed), do not have one. If the skill was already achieved, a warning message is returned and the simulated user passes on to attempting the next skill in the hierarchy or the next goal.
5. If the skill is a device skill and it was not already achieved, check that its conditions are satisfied. A device skill's conditions are considered as satisfied if their beliefs have equivalents, or not if negated, in belief memory. If any of the conditions of the device skill are not satisfied by any of the bindings, a warning message is returned and the simulated user will go back to point 2 as above. Otherwise, the device skill is considered as achievable given the binding.
6. If the skill is an external skill and it was not already achieved, consider it as achievable.
7. Mark the real percepts from the binding, which were required by the skill, as being remembered and used by it, and thus more difficult to forget. I will return to this point in the next section.
8. Replace the skill's percepts with the real percepts from the binding to create an intention, the instance of the skill.
9. If the intention is of a device skill, which has subskills, attempt each subskill in turn by returning to point 3.
10. If the intention was instead of an external skill:

- (a) Compute its time according to the probability distributions and chance of interruptions of the external skill.
 - (b) Mark percepts which were used as part of the intention as no longer in use or required to be remembered, such that the user could forget them quicker. I will return to this point in the next section.
 - (c) Add the beliefs from its effects to belief memory, as external skills are always considered as achieved.
 - (d) Perform a new conceptual inference to infer any new beliefs.
 - (e) Consider the intention (thus also external skill) as accomplished, and pass on to the next intention, skill or goal.
11. If the intention was instead for a device skill which had an action, send it to the device with the help of the controller.

Once the device updates its state, the simulated user begins a new cognitive cycle involving perception, the forgetting of old information (as I will explain in the next section) and conceptual inference. She will then check that the intention was achieved by checking that its effects are met in belief memory, as was explained above for device skills. If an intention is not achieved, a warning message is returned and the simulation run terminates. If the intention is achieved, the simulated user will reset the real percepts which had previously been marked as in use by the device skill in point 7, to indicate that they are no longer used, and could thus be forgotten quicker (more details below). She then removes the intention, as well as other achieved parent intentions, from the hierarchy, marking the real percepts that they used as no longer in use or required to be remembered, and passes to the next intention, skill or goal. The goal is considered as achieved when there are no more intentions left in the hierarchy.

6.4.2.5 Mechanisms for Remembrance/Forgetfulness

The COs who use the telehealthcare system described in Chapter 3, whom I have decided to use as inspiration for the modelling of examples for this thesis, rely on their memory for moving information between the applications which I have considered as part of the system. I have therefore decided to develop remembrance/forgetfulness mechanisms for the simulated user, which were not available in ICARUS. These mechanisms are simplistic yet reasonable, and based on the following assumptions or conventions:

- Humans have limited STM duration and capacity, and thus the simulated user should as well. In terms of duration, Atkinson and Shiffrin found that STM lasts for **15-30 seconds** ([37]). In terms of capacity, **the magic number 7 +/- 2 by Miller** ([157]), although largely contested, is still the most frequently cited in the literature. It suggests that a human can remember in STM as much as 7 +/- 2 items at once. Not to impose that such values be used for the simulated user's STM duration and capacity, the modeller can configure the user model with any values, and thus also have the option to change them to model humans with worse or better memory.
- A percept's attributes are better suited for being considered as items of memory than the percepts themselves. As some studies in psychology have shown that people can 'chunk' more or less information in one memory item depending on its spoken duration ([38]), sound similarity ([70]), familiarity ([184]) and meaning ([183, 50]), and attributes can consist of words and numbers of different sizes, I propose the reconsideration of the granularity of memory items as future work.
- Humans are likely to be more interested in remembering information from the system, as opposed to the system's structure, and therefore the simulated user remembers only those percepts and their attributes which hold information (e.g. referring to patients, measurements, etc., and not website tables, pages or buttons).
- As human reaction to contradictory beliefs about the system is unclear, as it may depend on her experience with the system and with computers, the simulated user will not remember beliefs. Instead, she will always re-infer them through conceptual inference.
- As the order between seeing new information and forgetting old one may depend on factors such as fatigue or attention, which are not modelled, the simulated user will always forget old information (i.e. information-type percept attributes) as an effect of seeing new one.
- Humans may well be more interested in remembering information that they use, or would later need to use, for their activities, than other information from the system. The simulated user will therefore remind herself of information which would make the skill that she is pursuing achievable, or that she would need for a later skill.

- Humans may well be slower at forgetting the information that they use, or would later need to use, for their activities, rather than that which they only see, but do not use. The simulated user will therefore also mark information that she uses for each skill, or would need for a later skill, and the modeller can input a higher duration (than for other information) until such information would be forgotten.
- Humans may well be quicker at forgetting information that they do no longer need for their activities, rather than that which they still need. The simulated user will reset information which is no longer in use for a skill and, if it is no longer in use for any other skill, she will forget it after the same time as for forgetting information which was never in use.

The main mechanism for remembrance/forgetfulness can be summarised as follows: whenever she perceives new percepts having attributes which refer to information (as explained in the third assumption from above), the simulated user considers that they are new, just remembered, percepts and attributes. Once an action is performed on the device, she considers that they become percepts/attributes which have been remembered for the duration of the action. After each subsequent action, if they are not re-perceived, the new duration of the action is added to their remembrance time. If they are re-perceived, the simulated user reminds herself of them and considers them again as new. Also, she reminds herself of individual attributes which are required for a skill, and of the percepts themselves, if the skill as using them becomes achievable, and marks as in use these attributes and the percepts which were defined without any attributes. Until this skill is achieved, they can only be forgotten if they exceed the maximum remembrance duration for used percepts/attributes (which should be higher than that for unused percepts/attributes). Once the skill is achieved, and/or they are not used by any (other) skills, they will be forgotten when their remembrance exceeds the normal maximum remembrance duration. The simulated user always forgets attributes first, and then the percepts themselves if they are not in use. After each perception step, the simulated user also forgets information-type attributes if her memory capacity for them has been exceeded, giving priority to those which are not used by skills, then to those which are used starting with the oldest remembered ones. Similarly, she only forgets percepts if they are left without attributes and are no longer in use.

An exception to the behaviour from above is the case in which the simulated user would need certain information from the device, for using it in a later skill. For example, a real user could need, as part of the skill of going to the Alerts page, to remember

certain alerts for the patient, for using them later in a skill of filling in alerts on the spreadsheet. For such cases, the modeller can set for the current skill for the user model, within the ‘percepts to remember’ component, the format of percepts which the simulated user should remind herself of and consider as in use, if she perceives them once the current skill is accomplished. If the simulated user will indeed find percepts respecting that format during perception, and mark them as required, they will only be forgotten if they exceed the maximum remembrance time for percepts/attributes which are in use. As this time should be longer than that for unused percepts/attributes, this makes it easier for her to remember them until they would be used by the skill requiring them. Also, when a skill requiring them is found, these percepts/attributes will have already been marked as in use for it.

As another exception to the regular remembrance/forgetfulness behaviour, the modeller can configure the user model such that the simulated user sets certain information-type percept attributes, or the percepts themselves, as ‘from external’ to indicate that they are available in her static environment, and therefore she would never like to forget them. This is useful, for example, for the case in which the modelled user writes information down on paper, information which would then be always available to her. The attributes which are set as ‘from external’ will never be considered as remembered, so the simulated user will never forget them due to having exceeded their remembrance time. Also, they are not considered when forgetting information because of the limited memory capacity. Their values may be updated, if a device action changes them from the system, which simulates the user also changing them in her environment.

I propose as future work the thorough evaluation of the mechanisms for remembrance/forgetfulness and their underlying assumptions or conventions, and their further refinement, with the help of a psychologist.

6.4.2.6 Behaviour of the Device Model

When it receives a requirement for carrying out an action from the simulated user, in the form of a user action, the device looks for an equivalent device action, which takes device state components which are equivalent to the percepts with which the user action was called. If such a device action is not found, it will return an error message. Otherwise, the device will try each identified device action in turn, by checking that its required device state components have real device state component equivalents from the device state. Once a device action thus instantiated is found, the device will compute its duration by using the probability distributions defined for the device action, and

the chance of interruptions. It will then change its state by following a procedure which must be defined by the modeller for that action, by adding/removing device state components, and return the new device state to the simulated user.

6.4.2.7 Behaviour of the Controller

The controller helps with the communication between the user and device models, being set by default to simulate a user's goal-directed behaviour on the device. It first initialises the models by using the inputs provided by the modeller. It then repeatedly requests the simulated user to do the next step for accomplishing her goals, sends actions required by the user (user actions) to the device, and returns the new device state to the user, requesting it to check the achievement of its latest intention. Any of these steps could result in an error message from the user or device model, in which case the run is terminated. The cycle finishes once the simulated user has achieved all of her goals.

As for my previous modelling approach, the controller also checks some mode confusion problems: whether user expectations from the effects of intentions having instantiated device skills are not met by her beliefs and, if so, whether the simulated user has not perceived well from the system. For the time being, these checks only validate modeller inputs. The former would only be possible if the simulated user does not check the effects of her intentions obtained from device skills, which may be reasonable for modelling a very experienced user. The latter would only be possible if a different perception behaviour is implemented for the user model, as the current one allows the simulated user to perceive everything which is on the screen and visible.

6.4.3 Inputs

The modeller must provide the following inputs to the new modelling approach:

- **For the user model:** the elements from LTM (the sets of concepts and skills) and the list of goals, a specification of repeatable goals, and the memory capacity and duration. The elements from LTM and the goals also appear as inputs in Icarus. The notion of repeatable goals was added by me to allow for several goals with the same name to be generated according to the provided workload (as will be described in the next chapter). I have also added the memory capacity and duration as motivated by the need for the remembrance and forgetfulness mechanisms. The elements from STM will be generated during the run of the models.

- **For the device model:** the initial device state (containing the initial set of structural-type device state components), the set of device actions, a specification of how the initial device state should be populated with information-type device state components, and a representation of the changes to the device state that would be triggered by each action during the run of the model. These inputs are similar to those for the device from PUM ([54, 46, 47]). I have additionally decided to consider the information from each device state as separate from its structure, in order to be able to populate the device state dynamically with information (as will be clarified in the next chapter).
- **For running the models:** a list of settings, either constants or probability distributions, defining the workload and other characteristics of the work environment, and a way of using this list to automatically generate the information which is needed by the two models during the simulation run. This category of inputs was added by me to allow for the automatic generation of information for the user and device models, according to characteristics of the work from the modelled deployment site.

Please note that other personal and contextual issues could have also been considered as inputs, because of their potential effects on efficiency. In particular, for a user, tiredness, physiological state, attention, focus level, and general personality traits such as seriousness and dedication to her work, could all influence the time that she spends on a computer system. Moreover, although I have considered interruptions (i.e. the user taking breaks from her work) in my modelling approach, the user could also be bothered in her work by other factors. For example, she could have a slower pace because of surrounding noise (e.g. if she works in a shared office, like in the case of the COs whom I have observed), the availability of someone to offer her support, or the fact that she is being observed. I propose the investigation of these and other factors, and the extension of the modelling approach to consider more factors which affect efficiency, as future work.

I will provide more details about the construction of the inputs, the means of collecting the information needed for them, and the emerging difficulties, in Chapters 7, 8 and 10.

Chapter 7

Engineering an Application for Running Simulations

7.1 Overview

In the previous chapter I have described the structure and behaviour of the new modelling approach. In this chapter, I will first describe the design and system requirements for the Java application in which I have implemented the modelling approach. I will then provide more details about the inputs in the form of model configurations and additional programming which are necessary for each simulation run. Finally, I will illustrate how the application can be used, by detailing how the different steps of the simulation are run individually and together.

7.2 Design of the Java Application Implementing the Modelling Approach

The structure of the application in which I have implemented the modelling approach is provided in Fig. 7.1. It shows the static relationship between the different classes of the application, as based on their attributes. The most important classes are the User and the Device, which are coordinated by the Controller. Most of the classes that the User class knows about should be familiar to the reader as corresponding to the constructs from the LTM and user state (including goals memory and STM) and their components, which were presented in the previous chapter. The Definition class is a helper class which defines a common basic structure for goals, user actions and beliefs, and which

is used by the classes corresponding to these notions. HeadandRem is another helper class which uses the basic structure from the Definition class and adds to it percepts to remember, forming a common structure to be used for calling skills and intentions. ObservantUser is a kind of User, the functionality of which includes the perception behaviour described in the previous chapter, and the cautious behaviour of checking that the effects of device skills/subskills have been met. Other classes can be developed as a 'kind of' User to develop different such behaviour. The relationships between the Device, DeviceState, DSComponent and DeviceAction- DSComponent were described the previous chapter. The TransitionManager class is responsible for transitions of the device state given an action. It knows about all of the device actions, and about device state components which would map with them.

I have omitted the classes which are concerned with running the application and computing or logging the results, as they do not belong to the conceptual structure of the modelling approach.

7.3 System Requirements for the Java Application

The modelling approach was implemented in Java under JDK (Java Development Kit) 1.6, by using the NetBeans 7.1.2 IDE (Integrated Development Environment) running under Windows 7 64 bit. I have also tested it under the latest version of Java. Its main functionality was explained in the previous chapter.

7.4 Modeller Inputs Required As Prerequisites for Running Simulations

As described in the previous chapter, the modeller must provide inputs for the user and device models, and for their interaction.

Table 7.1 describes where the inputs **for the user model** must be included, and their required format. For the description of LTM constructs and goals in XML (Extensible Markup Language) files, the modeller should decide on a certain level of detail, depending on the type of user work which is being modelled, and the purpose of the simulation. The specification of repeatable goals must indicate which goals are repeatable, and which collection of real percepts they can refer to. The value provided for STM duration should be set as higher for information that the simulated user would like to remember,

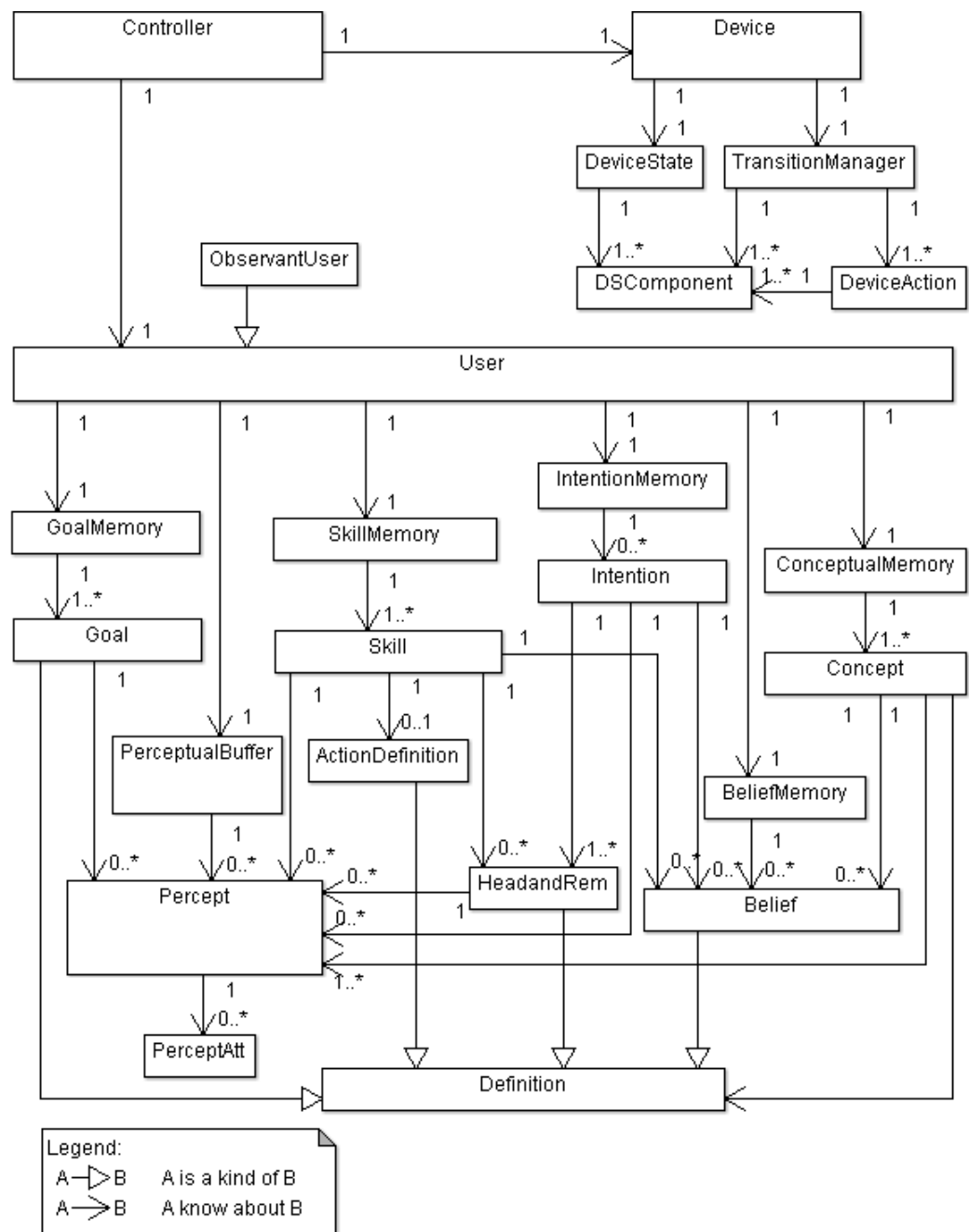


Figure 7.1: The Class Diagram

Input type	Location	Format
LTM constructs (concepts, device or external skills)	concepts.xml, device_skills.xml, ex- ternal_skills.xml	XML
Goals	goals.xml	XML
Specification of repeatable goals	repeatable_goals.txt	text
STM capacity	Configs class attributes	integer number
STM duration (for info to be remembered/not)	Configs class attributes	integer numbers

Table 7.1: Inputs required for the user model

as opposed to the one which she wouldn't.

Table 7.2 describes the **required inputs and formats for the device model**. The initial device state must only include structural device state components. The modeller must define within the Java implementation how the device is being populated with information-type device state components, and how device actions change the device state. Therefore, some basic experience with programming in Java is required.

More detailed requirements for the user and device model inputs are provided in Appendix F.

In my examples, I have grouped the files required for each example in a separate folder from the main Java project folder. At the top of my current XML input files, I have included for validation purposes and guidance a DTD which defines the compulsory or optional components to be included for each notion.

I propose the automatic generation, outside of the Java implementation of the modelling approach, of the **inputs required for each run of the models**. The modeller should provide a *list of settings* including constants or probability distributions for the workload or different characteristics of the work environment. A script or program written in her choice of programming language could use the list to automatically generate the inputs for the models. The modeller should also develop a way of running the script or program repeatedly, for obtaining model inputs for repeated simulation runs. I have provided a list of settings and a script for the examples described in this thesis, as I will detail in the next section, but they would need to be changed or replaced for other examples.

Input type	Location	Format
Initial device state (only with structural device state components)	current_device_state.xml	XML
Device actions	actions.xml	XML
How to populate initial device state with information	DeviceState class, populateInfo method	Java code
Changes of the device state triggered by actions	TransitionManager class, computeDeviceEnd method	Java code

Table 7.2: Inputs required for the device model

7.5 How it All Works

My modelling approach can be used for performing repeated simulations of the user working on the device and in her environment, by following the steps provided in Fig. 7.2.

Before each simulation run, the list of configurations which was provided by the modeller is used to automatically generate inputs for the run.

The simulation run uses these inputs, as well as those which were provided for the user and device model, as described in Section 7.4. During it, the user and device model are run in parallel, simulating a user working on the system and doing any actions external to it within a deployment. Their communication is coordinated by the controller. The outputs of the run are logs containing the total duration of the user's work on the system, the durations for intermediary goals and intentions, and details about the steps performed by the simulated user and device.

When performing repeated runs, the mean and standard deviation of their total times, and a histogram containing the relative frequency distribution (term explained in [3]) of these times up to the current run (i.e. the percentage of runs in which the total time falls within one of several equal time intervals) are computed. A high enough number of runs will lead to a stable distribution, predicting the efficiency of the system and work process within the modelled deployment.

The modeller can change the inputs to the user model, device model or workload, to reflect changes in the user skills, way of doing things, system design or workload

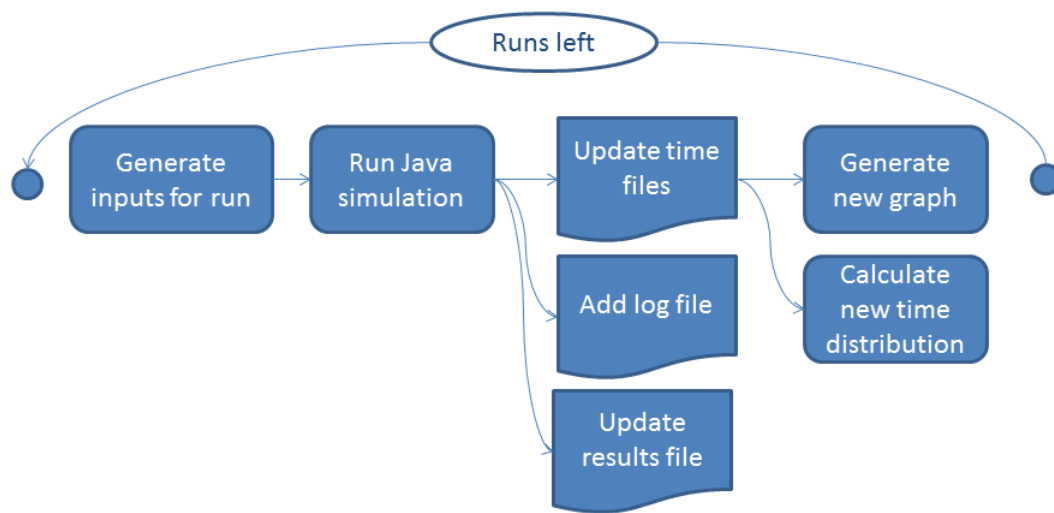


Figure 7.2: Process for Performing Repeated Runs

characteristics within a potential, real or hypothetical, deployment. These changes can interact with each other in a way which makes it difficult for the naked eye to foresee efficiency there. On performing another high enough number of runs for each potential deployment, the modeller can obtain a new stable distribution predicting the efficiency of the system and work process within that deployment. This result, together with the investigation of the log data on the steps performed by the user, can help analyse the effects of changes on efficiency, as I will show in Chapter 9.

In the following subsections I will describe my implementation choices for each of these steps.

7.5.1 Generating the Inputs Required for the Simulation Run

For the examples provided in this thesis, which will be described in Chapter 9, I have created a Perl script which takes a list of settings and creates XML files to be used as inputs by the models. The list of settings (from *'workload.txt'*) are represented as constants, percentages, means and standard deviations, or bounds of the values for different workload characteristics: the total number of patients, the number of patients having each type of alert, etc. The script generates random numbers from distributions, or directly uses the constants from the list of settings, to automatically create XML files which will act as inputs for a simulation run.

7.5.2 Running the Java Simulation

When the application is launched, the controller commences the initialisation process for the device and, afterwards, for the user.

The device first initialises the structural device state components of its initial device state and its actions, by parsing the input XML files for the device model described in section 7.4. The device will also use the XML files which were automatically generated as inputs for the run in the previous step to populate the information-type device state components of the initial device state, according to the functionality defined by the modeller (as described in Section 7.4). These XML files will therefore act as a simplified database for the device.

If there are any problems during the initialisation of the device, error messages are returned to the modeller and the processing stops. If the device initialisation was successful, the controller will then require the simulated user to initialise its components. To do this, the user first initialises her LTM constructs and goals by parsing the input XML files for them described in section 7.4. For repeatable goals, she uses the *'repeatable_goals.txt'* input file to identify the percepts on which they should apply and add the appropriate goals to her list of goals. Once LTM constructs and goals are initialised, the simulated user also initialises her percepts and beliefs. For the former, she perceives the initial device state sent to it by the controller. For the latter, she performs conceptual inference to instantiate concepts into beliefs by using the new percepts from the perceptual buffer. As the simulated user will mostly work with information-type percepts, the inputs which were automatically generated in the input XML files used for the runs, and which were used to populate the information-type device state components, will constitute her workload.

As for the initialisation of the device, any problems with the initialisation of the user lead to the halting of the application. If the user initialisation is also successful, the controller launches the skill execution behaviour of the simulated user. This behaviour was described in Chapter 6.

During its run, the application keeps track of the time it takes for the simulated user to accomplish each of her intentions, goals, and the total time of her work. This information also constitutes outputs for the run, which are logged in the *'times_intentions.log'*, *'times_goals.log'* and *'times_total.log'* files from the main Java project folder. For each new simulation run, as the times are computed, the application appends them to the log files. Another important output of the application is a summary description of the

intentions and goals which were achieved by the simulated user, together with their duration and any warnings which appeared. This description is logged for each run by appending conclusions on achieved intentions/goals, as they take place in the simulation, to the *'results.txt'* file from the main project folder.

If the modeller has chosen to activate full logging, the application will also log the detailed description of the steps performed by the simulated user and device for each run within a different text file from the *'Logged.data'* folder of the main Java project folder.

7.5.3 Generating Time Distribution Charts

I have developed a script in R, which can parse the results from the total time log file after any one run and generate a histogram representing their relative frequency distribution: the percentage of runs the total time of which falls into each of a number of equal time intervals. On the X axis we have equal time intervals starting with the interval in which the minimum total time of the runs would fall. On the Y axis, we have the percentages of runs. It saves the histogram as a new pdf file in the *'Graphs\Histograms\'* folder of the NetBeans project folder.

7.5.4 Calculating Time Distributions

The total time log output by a new run is also used to calculate the means and sample standard deviations (according to the formula taken from [4]) of the total times up to that run¹. I have developed a Perl script for this purpose. It appends the calculated results to the text file *'total_time_distributions.txt'* from the main NetBeans project folder.

7.5.5 Putting It All Together

I have described each of the steps involved for any one run, but how can we do repeated runs?

I have developed a Windows batch file (*'Runner.bat'*) which acts as the starting point for the whole process. For each simulation run, it initialises the log files and requires from the modeller inputs about which example to run and whether logging of detailed steps for the user and device is to be activated. It then calls a number of

¹I use sample standard deviations as the total times obtained from the runs are only samples of the whole population of total time values

other Windows batch files, which are responsible for performing the main steps of the process:

1. Calling the Perl script which automatically generates inputs for the run out of the list of configurations
2. Launching the Java application, with the inputs taken from the modeller which were mentioned above, to obtain its time and summary/ detailed step logs
3. Calling the Perl script to compute and save the mean and standard deviation of the times up to the current run
4. Calling the R script to generate a new histogram representing the relative frequency distribution of the total times up to the current run

Chapter 8

Data Collection from a Case Study of a Telehealthcare System in Use

8.1 Overview

We have seen in the previous chapter how the inputs for the models would need to be provided. As I described for the methodology in Chapter 4, the data for these inputs should be collected from the reference deployment site for building the initial models. Then, for each potential deployment site, information about how it differs from the reference deployment site would need to be collected. This would help to build a model for the potential deployment site, which could be used to predict efficiency there. For hypothetical deployment sites, the modeller could explore the effects on efficiency of any change to the data.

In the next chapter I will provide a series of examples in which I modelled hypothetical scenarios which could appear for the same or different deployments, to show how my methodology and modelling approach could be used in practice. I collected the data needed for the inputs of the models during my usability study of the telehealthcare system used in Lothian, which was described in Chapter 3. In this chapter, I will briefly explain how I collected and analysed this data, and how I decided on the inputs for building the first examples.

8.2 Purposes of the Study for My Modelling Approach

Apart from the aim of collecting information about user perception of the system's usability, my study of the telehealthcare system used in Lothian was also intended to

help me collect data and qualitative observations to use as inputs for building realistic hypothetical user and device models for my examples. The following were the questions that I addressed for meeting this aim:

- For instantiating user models:
 - For building the device skills: What is the order in which users perform tasks on the system, when and how frequently? What are the different strategies used by users for the same task? In what way does this vary according to the users' experience?
 - For building the external skills: What additional tools or systems from their environment do users use? When do they use them? How much time do they usually spend using them?
 - For building the concepts: In what way do users analyse the information provided to them on the system (e.g. to decide on the patient's state)? What constitutes a reason for their device or external skills?
 - For goals, I considered that the monitoring of patients is the simulated users' goal.
 - For STM capacity and duration: When do users forget information? How is this influenced by the time that it takes them to perform external skills, or the duration of actions on the system?
- For instantiating the device model: What does the state where the users start in look like? What information does it contain? What actions does the system have available, and which ones are used by the users? How do the actions change the system's structure and information content? How long do the actions take on the system (i.e. user waiting time)?
- For running the models: What is the normal workload of the users, in terms of number of patients, their alerts, their availability when telephoned, their answers on the phone? How can this workload vary?

I should add that the results on user perceptions of usability were also intended to inform and motivate my modelling of simulated users.

8.3 Data Collection Methods

The data collection methods that I employed for the first aim of the study, which I described in Chapter 3, were also used for collecting data for my models. In particular, I used the **videos that I recorded of the computer screen while users were working** to collect information about the order and duration of user tasks, and for helping me understand user strategies and differences between users. Even if I only filmed the computer screen, **I also recorded the sound of the videos** in order to help me deduce better the timing of user tasks, and the external actions that users performed in their environment. The data collected through video recording was complemented by the data which I obtained from **interviews**. During them, I always asked users to describe the order and reasons for their actions, what other tools or systems they used and why, and how they would work differently under certain conditions.

With regard to the system, the videos were also intended as a means of collecting information about the different states, available actions and how the states change. Throughout the duration of the study I also had **access to a test website for the system, to two patient tablets, and to the spreadsheet which would be filled in by COs and sent to the care providers**. This allowed me to get some hands-on experience with the system, even if the test website only included test data.

8.4 Data Analysis Methods

I used the Anvil video annotation research tool ([130]) to time and annotate the videos. It helped me distinguish, name (i.e. annotate), and create a hierarchy of different user actions, to be represented as skills and subskills for the user model. I observed that some user actions were part of clear strategies for certain tasks, while others were only carried out under certain circumstances. I could also add comments to the different actions, which helped me explain any observed reasons behind them, or mark the actions during which the user was interrupted by a colleague or was taking a break. Most importantly, Anvil helped me time the different user actions which I had delimited from the videos. Its use ensured a higher precision of the times than I could have obtained by using a simple media player or by timing the videos manually using a stop-watch. Anvil allowed me to export the annotated actions from each level in their hierarchy, with their durations, in a text file. I populated a Microsoft Excel 2010 spreadsheet with the durations for each action from this file, and I then computed the means and standard deviations of

these durations. These results were then used to populate the time components of device actions and external skills.

When annotating and timing the videos, I considered that the user's movement for doing an operation on the system (e.g. click), as well as the carrying out of this operation by the system until its state changed, were all part of one user action on the system. As the real user's hands were not filmed on my videos, and therefore their complete movement was not clear, I considered that this would be a reasonable simplification. The times used for the time components of device actions therefore include the real user's movement time, and the actual waiting time for the device to change its state, as also mentioned in Chapter 6.

By comparing the results obtained for different users, I could understand differences between them in terms of the actions that they took, the strategies used for different tasks, the time that they spent doing certain actions, and the external tools that they used.

I only made full annotations for two users- a beginner and a more experienced CO, whom I decided to model, as I will explain in section 8.6. I also annotated some actions of other two COs, as I decided to complement the user models with some additional actions and different strategies, as I will explain at the end of this chapter.

For modelling purposes, I also performed simple observations of the videos, and made notes about them. This helped me to collect information about the workload of the users during the observations (i.e. how many patients they monitored, what alerts each of them triggered, what were the outcomes for each) or to draw higher level conclusions about differences between users.

I analysed the interview transcripts by using the template analysis approach ([22]), as was described in Chapter 3. User explanations about their sequence of actions (strategies) for monitoring patients, why they did certain actions, at what information they looked at, and what external tools they used and why complemented my results from the video annotations.

For modelling device states and how they change, I took notes of observations both from the videos and from the test website and spreadsheet.

My observations, together with the results on user perceptions of the usability of the system, also led to conclusions about the areas of the system which caused inefficiencies and frustrated users. The exploration of the effects on efficiency of potential solutions to these areas is the subject of two of my example scenarios, which I will present in the next chapter.

8.5 Data Collection Limitations

The data collection methods that I used for this part of the study were unfortunately restricted by the permissions that I received from the NHS. As for the first aim of the study, it would have been very useful to have the possibility of collecting interaction logs either from the software company implementing the system, or from my own application instrumenting the users' browsers. Interaction logs could have provided me with information on user strategies in using the system, differences between users and what poses problems, which would have been useful for building the user model. They would have also been better sources of information, than my practising on the test system, for building the main constructs from the system model. Most importantly, they could have contained timing data for the different device actions, which would have been much more numerous and accurate than that collected and extracted by me from video observations of the computer screen. As also explained in Chapter 3, collecting interaction logs was unfortunately impossible, and observations of the screen were the only solution.

It would have also been extremely useful to be able to film the participants and their work environment during observations. This would have allowed me to better capture actions which are done by the users in their environment or by using external tools, and their duration. I needed to take written notes of all external actions during observations, map them with the videos to deduce their order, and listen to the videos to get a rough estimate of their durations. The video recording of the users would have also allowed me to separately capture user movement time for performing an action on the system, which I needed to include within the duration of device actions in the current setup. My video observation of the users was unfortunately not possible due to NHS restrictions. In particular, throughout the duration of the study I held an observer contract with the NHS, which only allowed me to take video recordings of computer screens. It took me two months to obtain this contract, and a further delay for obtaining an ethical approval for video recording NHS staff was unacceptable for my collaborators for the purposes of the usability evaluation.

Apart from the data collection methods used, and despite succeeding to recruit the great majority of system users for the study (as was described in Chapter 3), the data that I obtained was also negatively influenced by the number of participants, their frequency of use of the system's features and their fluctuating workload. In particular, the project was still at a relatively small scale, so the total number of users occupying

each role was small. Secondly, while I managed to observe (and video record) all of the NHS 24 COs as given by their approval, the same cannot be said about care providers. Although they all kindly agreed to my observation, some of them did not have any patients to monitor that day (i.e. there were no emails from the COs). Moreover, even if they had patients to monitor, this normally only lasted for 10-15 minutes due to their small number. Thirdly, even for participants whom I did observe for the whole hour, not all of them would use the same system features or similarly frequently, so I did not always get enough data on their timing.

8.6 Choice of Users to Inspire User Models

Due to the fact that I had much more data from video-recorded observations for COs than for care providers (as explained in the previous section), I decided to use as inspiration for the models from my examples observations of NHS 24 COs. For each of them, I had one hour worth of video-recorded data, and interviews.

It is often the case that users who are less experienced with a computer system and work process spend more time in their work than experienced ones. Experience with the system should therefore be an important consideration for a healthcare organisation in deciding on the time to allocate to COs for monitoring work. As explained in Chapter 3, COs varied from beginners (but already used to the main steps that they had to perform on the system) to experienced. Moreover, during my study I found that, despite having been formally trained on following certain steps in their monitoring work, COs adapted these steps such that they were more comfortable to them. This was also supported by the videos, where I observed that each CO worked differently, only adopting similar strategies on some types of tasks. This may have been caused by their experience, but also by personal preference. Their different ways of working, i.e. different strategies for different tasks, also led to different times that they would spend in monitoring patients each day. I therefore believe that a consideration of the strategies employed for working is also important in an analysis of the time required by COs in their monitoring work.

Out of these considerations, I chose user experience, in terms of the time spent on individual tasks, and strategies for work as two criteria on the basis of which my modelling approach could be used to predict efficiency. With regard to the different strategies, I decided to concentrate on a task which is of critical importance for the monitoring work of COs: the filling in of alerts from the website's Alerts page onto the spreadsheet. More exactly, COs need to answer with 'yes'/'no' to questions from the

spreadsheet regarding the existence of each type of alert for the patient on the website. On watching the videos, I found that COs used one of three different strategies for this task, for which they relied on their memory in different ways. I will describe the three strategies in the next chapter.

I chose as the main inspiration for my user models two users who were consistent and clear in the way that they were doing tasks, as observed from the videos and explained during their interviews. The first user, who was experienced, was normally much quicker in her actions on the system than the second user, who was a beginner. The two users used different strategies for filling in alerts onto the spreadsheet.

8.7 Building Inputs for the User and Device Models

I built two initial user models which were each mostly inspired from each of the two real users. To be able to exemplify how my approach could be used to predict the effect on efficiency of user experience and of the strategy used for filling in alerts on the spreadsheet, it was necessary for the user models to be similar in other respects. In particular, I decided to:

- Keep a common workload for the two user models: To best reflect the work that was performed by all of the observed users, I chose as workload for the simulated users the maximum number of patients having each possible combination of alerts which I observed from the videos. In the list of settings used for generating the workload, available in Appendix G, I included the total number of patients for each of the two conditions- COPD and CHF, the number of patients having each combination of alerts for these conditions (the different paths), information about them picking up the phone and possible answers to the questions asked by COs.
- Other than the different strategies for filling in alerts on the spreadsheet, implement common work for the two user models: I implemented for each user model the real user's steps of the work (in terms of device skills, external skills with their time, and concepts) which concerned the strategy for filling in alerts on the spreadsheet. The strategies were therefore different for the two user models. For the rest of the work, when the real users had common steps, I used them to build the associated skills and concepts for the user models. When the steps of the real users were different because the users used different strategies for those parts of their work, I chose to implement for both user models the steps performed by

one of the real users. Due to the decisions regarding a common workload from above, I also had to implement some additional steps for each of the user models. Where these steps were only performed by one of the real users, I used them for both of the user models. Where none of the real users performed these steps (they were only necessary due to the decision regarding workload), I used instead the steps of another observed CO for both of the user models. Both of these decisions made the simulated users equal for these steps.

For the initial device state and the functionality about how this state changes, I only represented as device state components those real system elements which are used and required for routine monitoring work by the users. For device actions, I represented the operations which the users use from the system in their monitoring work.

Tables 8.1 and 8.2 include the times that I have used for each device action or external skill. I have indicated for each which real user it was taken from, and the value which I have chosen for its lower bound. I only considered times without interruptions from the real user observations for each device action/external skill, and I ignored breaks that the real users took in the videos. I thus modelled ideal work conditions. Some of the means and standard deviations from the tables are unrealistic, because those particular user actions (i.e. use of system features) had been seldom performed by the real user in my observation. To partially deal with this issue, I used lower bounds to limit the minimum value of the random numbers generated from each of the means and standard deviations. I chose as the value of the lower bounds the minimum times that I observed the real user perform those actions on the device or in her environment, if she had repeated them a reasonable number of times, or invented, but realistic times reflecting what would be humanly possible. Deciding on these invented times involved my judgement of whether the user was already very quick in doing the actions on the device or in her environment from the video, and sometimes even timing myself doing them by using a stop watch.

For both of the user models, I used an invented time for the device action of scrolling down within a website's page. I did not observe this action from the videos, because the real users did not need to scroll down to retrieve patients during my observation, but it was necessary for modelling the monitoring of the larger number of patients constituting the chosen workload. I made the convention that users use the strategy of doing repeated clicks on the arrow of the scroll bar, and that it takes them the time of a click from KLM ([61])- 400 ms, plus 200 ms (reasonable invented time) for the device to move the page in response. With regard to the duration of the external skill of

waiting on the phone for the patient to pick up, I used for it the maximum time that user 2 would wait until hanging up (user 1 was never in this situation). As waiting on the phone is influenced by the patient, I considered the percentage of patients picking up, and the probability distribution of the time in which they do, within the characterisation of the workload (which is used to automatically generate the inputs required for each simulation run). Therefore, and as an exception, the duration of the external skill of waiting on the phone for a patient to pick up is computed either from the patient's time distribution, if the patient picks up in the inputs, or by using the distribution from the time components of the external skill, if she doesn't.

Device action/external skill	Mean	SD	Taken from	Lower bound
Scroll_down_homepage	700	350	invented	600
Open_patient	2418.57143	1244.64759	U1	600
Click_new_s	3225	862.43	U1	2000
Click_website/spreadsheet_tab	666.477273	563.766153	U1	80 (min)
Click_badge from homepage	7653.75	2105.9639	U1	5000 (min)
Click_badge from Alerts	8500	707.106781	another user	6025.126267
Click_alerts from badge	4994.28571	848.957344	U1	3500
Click_alerts from PM	4472.5	591.741779	U1	3000
Click_alerts from surveys	4472.5	591.741779	U1	3000
Click_PM from Alerts	3432.5	768.130848	U1	2500
Click_surveys from Alerts	5303.33333	1035.58357	U1	4000
Copy_ID	2726.25	506.949913	U1	2000
Paste_ID	1531.42857	730.808424	U1	750
Copy_phone_number	2847.5	481.330003	U1	2200
Paste_phone_number	2232.85714	195.253193	U1	750
Enter_time	2308.75	285.779011	U1	1500
Write_name	4607.14286	937.29601	U1	3000
Select_cp_and_condition	4658.33333	1543.05433	U1	3300
Click_s_pg_2	1188.33333	474.65426	U1	500
Click_homepage	1973.75	688.765718	U1	1000
Select_answer	2403.33333	869.239018	U1	1540
Select_answer_phone_question	5030	28.284271	U1	4000
Select_answer_retaking	7930	0	U1	
Enter_call_headers	19730	4780.04184	U2	5000
Enter_call_headers_no_reply	22700	0	U2	
Select_headers	7572.5	1781.60241	U1	5000
Write_note	38342.5	13265.1625	U1	10000
Write_notes	38342.5	13265.1625	U1	10000
Write_reason	6000	1414.21356	another user	4000
Open_alert	1795	480.860607	U1	1000
Select_headers_for_alert	2878	349.289074	U1	2400
Write_comments_on_alert	11220.625	9542.2188	U1	3000
Save_s	13595.7143	1989.31192	U1	10500
Close_s	3887.14286	1539.6289	U1	1500
Retrieve_name	1000	763.675324	U1	400
Retrieve_cp_and_condition	1000	763.675324	U1	400
Retrieve_alerts	6181.66667	4997.70514	U1	1000
Retrieve_alerts2	1920	1767.76695	U1	500
Retrieve_alerts_for_notes	2880	1241.45076	U1	1000
Retrieve_alerts_PM	14430	0	U1	
Retrieve_alerts_surveys	1236.66667	354.729944	U1	900
Retrieve_notes	22450	0	U2	
Prepare_documentation	9470	0	U1	
Dial	11140	0	U1	
Wait_on_phone	30000	0	convention	
Say_intro	8510	0	U1	
Say_ending	17020	0	U1	
Email_cp	57796.6666	12475.7458	U1	40000
Retake_measurements	144260	0	U1	
Write_for_stats	5080	1409.9409	U1	3000

Table 8.1: Time taken for device actions/external skills for user 1

Device action/external skill	Mean	SD	Taken from	Lower bound
Scroll_down_homepage	700	350	invented	600
Open_patient	4923.33333	5066.580043	U2	600
Click_new_s	3756.66667	1672.493149	U2	1500
Click_website/spreadsheet_tab	1581.77778	1057.035496	U2	170 (min)
Click_badge from homepage	12500	2948.372432	U2	6000
Click_badge from Alerts	8500	707.106781	another user	6025.126267
Click_alerts from badge	15870	188.7975106	U2	4500
Click_alerts from PM	12863.3333	46.188022	U2	4000
Click_alerts from surveys	12863.3333	46.188022	U2	4000
Click_PM from Alerts	11850	469.254728	U2	2500
Click_surveys from Alerts	15170	438.406204	U2	4000
Copy_ID	3173.33333	575.007246	U2	2000
Paste_ID	2393.33333	266.520793	U2	1500
Copy_phone_number	3673.33333	798.268961	U2	2200
Paste_phone_number	3060	351.567917	U2	1500
Enter_time	2670	425.675933	U2	1500
Write_name	4103.33333	448.590385	U2	3000
Select_cp_and_condition	5756.66667	375.011111	U2	3300
Click_s_pg_2	1750	290	U2	500
Click_homepage	6785	1736.349619	U2	1000
Select_answer	4282.22222	2313.524248	U2	1540
Select_answer_phone_question	8720	5162.317309	U2	4000
Select_answer_retaking	4630	0	U2	
Enter_call_headers	19730	4780.041841	U2	5000
Enter_call_headers_no_reply	22700	0	U2	5000
Select_headers	15423.3333	3822.333493	U2	5000
Write_note	34236.6667	10977.64243	U2	10000
Write_notes	34236.6667	10977.64243	U2	10000
Write_reason	6000	1414.213562	another user	4000
Open_alert	4358	610.999182	U2	1300
Select_headers_for_alert	9778	5814.591129	U2	2400
Write_comments_on_alert	27135	2566.797616	U2	10000
Save_s	44460	12092.2992	U2	10500
Close_s	3695	2920.351006	U2	1500
Retrieve_name	1000	763.675324	U1	400
Retrieve_cp_and_condition	1000	763.675324	U1	400
Retrieve question (only for U2)	3251.42857	1043.622355	U2	1000
Retrieve_alerts	14275	7330.903764	U2	1000
Retrieve_alerts_for_notes	7970	763.675324	U2	1000
Retrieve_alerts_PM	3275	2100.10714	U2	900
Retrieve_alerts_surveys	2190	438.406204	U2	900
Retrieve_notes	22450	0	U2	
Prepare_documentation	19110	0	U2	
Dial	16795	3924.442636	U2	10000
Wait_on_phone	30000	0	convention	
Say_intro	14415	1859.690835	U2	8000
Say_ending	30790	16196.20635	U2	10000
Email_cp	117255	8464.068171	U2	40000
Retake_measurements	79350	0	U2	
Write_for_stats	5080	1409.940897	U1	3000

Table 8.2: Time taken for device actions/external skills for user 2

Chapter 9

Evaluation of the Methodology and Modelling Approach

9.1 Overview

Let us suppose that, within an existing telehealthcare deployment in a call centre, management would like to analyse and look for solutions for improving the efficiency of the system and work process surrounding it. They are interested in finding answers to the following general problems:

- What should be the optimal time to offer to COs for their work?
- What kind of training would be useful to help improve the efficiency of the system and work process?
- Is investment in hardware worth it from the point of view of system and work process efficiency?
- Is investment in making certain changes to system design worth it from the point of view of system and work process efficiency?
- Would COs manage to cope with a considerably increased workload in the time that could be offered to them in case of an epidemic?

In this chapter, I will demonstrate how my methodology and modelling approach could be used for seeking answers for such problems. I will first describe the general work that is being performed in the deployment within a basic case study. Afterwards, I will consider 4 scenarios addressing each of the problems.

9.2 Basic Case Study

Let us assume that COs within the deployment site work with a telehealthcare system which is similar to the one described in Chapter 3. They have a **total time of 2h 30min** to monitor patients every day, and any changes to this time should result in one which is a multiple of 5 minutes. We are modelling a typical day within a certain month, in which they are monitoring **17 COPD patients, 8 highlighted as red and 9 as orange, and 3 CHF patients, 2 highlighted as red and 1 as orange**. The deployment has a small scale, with a total number of 30 COPD and 10 CHF patients benefiting from the service.

The main steps of monitoring a patient, as inspired by my observations of the deployment described in Chapter 3 (please refer to the description of the telemonitoring website from there), involve the user (CO):

1. *Scrolling down to retrieve the next patient who is highlighted in red or orange from the table of patients from the homepage, if necessary (if she is not already rendered on the page).*
2. *Opening a new spreadsheet.*
3. *Filling in information about the patient on the first page of the spreadsheet by copying or remembering it from the patient's badge.*
4. *Replying to the questions regarding alerts from the second page of the spreadsheet, by retrieving the alerts from the patient's Alerts page, or from other pages of the website. The user can use one of three strategies for filling in the spreadsheet with alert information from the Alerts page:*
 - **Strategy 1:** *Try to remember all of the alerts from the Alerts page, and then answer to all of the questions from the spreadsheet about them. If the user forgets the answer to a question, she returns briefly to the Alerts page to double check it.*
 - **Strategy 2:** *Fill in one question at a time, by looking at the question, then retrieving the answer from the website's Alerts page, and then returning to the spreadsheet to answer to the question.*
 - **Strategy 3:** *Use paper to write down all of the alerts from the Alerts page, and then to fill in the answer for each question from the spreadsheet.*

5. *If the spreadsheet suggests it, telephoning the patient and filling in her replies to the questions from the spreadsheet.*
6. *Filling in the outcome on the patient's state on the spreadsheet, as suggested by it, by filling in the headers and entering comments about the patient's alerts and a summary of the discussion with the patient (if there was one).*
7. *Saving the spreadsheet.*
8. *Sending an email to the care provider, if this was suggested by the outcome from the spreadsheet.*
9. *Closing each of the open alerts for the patient from the Alerts page, by selecting headers and entering notes (the same as the comments from the spreadsheet) for them.*
10. *Writing down on paper conclusions about the patient.*
11. *Returning to the homepage and to the patient's position in the table, as the user knows that she must repeat the process for all of the patients who are highlighted in red or orange and follow the current patient the table.*

I must add here that for each of these steps the user will need to move in between the website and spreadsheet tab and their different pages.

Let us further assume that **within the deployment site we have two users- one experienced (user 1) and one beginner (user 2)** who are representative regarding the time that they spend in doing their actions for their level of experience. The two users work the same way, apart from using different strategies for filling in alerts on the spreadsheet: **user 1 uses strategy 1 (from above), and user 2 uses strategy 2.** We assume that the users' STM is characterised by a capacity of maximum 7 memory items (the average number from Miller's magic number [157]), a maximum remembrance time of 22.5 seconds for memory items which the user wants to remember (the average number from Atkinson and Shiffrin ([37]), and of 5.5 seconds for those that she doesn't (an average observed from videos for the real users).

9.3 First Scenario- Effects of User Experience and Ways of Working

In this scenario I will exemplify how management could address the first and second problem which were raised in the overview of this chapter. Let us suppose that they would like to consider *user experience*, in terms of the time which they spend in doing their tasks, and *the strategy that they use for filling in alerts onto the spreadsheet*, as the criteria in their analysis. In this case, they could break the problems down into the following more specific questions:

- Question 1: What is the current optimal time to offer to users with different levels of experience (can the allocated time be reduced to help save resources, or would they need more time)?
- Question 2: What would be the optimal time to offer to users with different levels of experience if they switch to using a different strategy?
- Question 3: How would the optimal time vary between users with different levels of experience if they use the same strategy?
- Question 4: Is there a strategy which would result in better system and work process efficiency, for each level of experience, which could be used for training purposes?
- Question 5: Is there a strategy which would result in better efficiency overall, which could be used for training purposes?

For answering the first question, I propose modelling the two initial representative users, and running simulations for each of them. For answering the rest of the questions, we can change the user models to represent the users' hypothetical change of strategy for filling in alerts onto the spreadsheet, keep the device model as the same, and run new simulations. I will address each of the questions in turn.

9.3.1 Addressing Question 1: Analysis of the Current Context

The list of configurations characterising user workload and, in a simplified way, the system's database, is provided in Appendix G. Different paths stand for different combinations of alerts, and so the configurations the name of which includes the word

”path” refer to the numbers of patients having each of the possible combinations of alerts.

The inputs in terms of time for device actions and external skills for the initial user models were provided, as inspired from two real COs, in the previous chapter. The users are modelled the same way, apart from their two strategies for filling in alerts on the spreadsheet, which can be summarised as follows:

- Strategy 1: once on the second page of the spreadsheet, user 1 opens the website tab, goes to the patient’s Alerts page, sets herself to remember all of the alerts, or their absence, from there (as percepts to remember, which means that they will be harder to forget), returns to the spreadsheet tab and fills in each question in turn with ‘yes’ if it refers to an existing alert, and ‘no’ if not. If she forgets the alert or its absence which would answer a question, and knows that it would be present on the Alerts page, she quickly goes back to the website tab to remember it (as a percept to remember), and returns to the spreadsheet tab to fill it in. I have not considered as an external skill, and thus allocated time, for looking at a question before filling in its answer or before user 1 realises she forgot the alert, or its absence, which would help answer it, because this time was not clear from the videos of the real user. The real user was very quick and either knew the questions from experience, or read each while answering the previous question.
- Strategy 2: once on the second page of the spreadsheet, user 2 retrieves (i.e. looks at) the next question to answer, and if she knows that the answer is on the Alerts page, she opens the website tab, goes to the Alerts page (if necessary), remembers the alert corresponding to the question or its absence (in percepts to remember), returns to the spreadsheet tab and fills it in on the spreadsheet. She then proceeds to the next question. By the time she is at the next question, user 2 could have forgotten the alert, or its absence, which would help answer it, because she had not set herself to remember other information than that which helped answer the previous question. This strategy will therefore include more movement between tabs. I have considered as an external skill, with allocated time, the retrieval of a question (i.e. looking at it) before searching for the alert, or its absence, helping to answer it for user 2, because this time was clear from the video of the real user. As for user 1, I have not instead considered for her the time of looking at the question again before entering the answer, because it was not clear from the video. The real user might have not read the question again, or only read it while

filling in its answer.

For both of these cases, if the user knows that an alert could be present on other pages of the website than the Alerts page, she goes there to retrieve it.

I ran the models repeatedly until the results from the generated histograms were stable enough. As a convention, I considered that **results are stable if the percentages of runs from each time interval do not vary by more than 1% over 50 consecutive runs.**

The obtained histograms are presented in Figures 9.1 (for user 1) and 9.2 (for user 2). They give us an indication of the total time that an experienced and beginner user would spend monitoring the 20 patients if they use their current strategy. What is clear from these histograms is that user 1 (the experienced user) would spend less time than user 2 (the beginner). More exactly, the difference of their modal times (most frequent total times) is 1h 1min 40s.

Looking only at the histogram for the first user, her maximum time is 1h 23min 20s, which is far from the 2h 30min which are allocated to users. This means that the time for an experienced user like herself could be reduced, and thus save resources. The safest would be to offer her 1h 25min. If we were to offer her 1h 20min, she could exceed this time in 12.67% of runs. This may prove unsafe for the monitored patients, but at the same time it would be more economical, and thus considerations of costs and risks are important for deciding on time allocation.

The second user needs a total time of up to 2h 27min 30s, which means that she can barely monitor the 20 patients in the total 2h 30min that she has allocated and, moreover, that she will surely not be able to monitor more patients in this time. The safest option would be to keep as allocated for her the full 2h 30min. Nevertheless, as only in 5.16% of runs she would exceed 2h 25min, the management could decide to offer her this time if it is deemed safe enough for the patients.

9.3.2 Addressing Question 2: Changing the Strategy of the Users

The strategy used by the two users whom we have modelled for filling in alerts onto the spreadsheet may not result in the best efficiency given their level of experience. We could change the initial models to simulate the users using the other existing strategies, and investigate which strategy would be best for them.

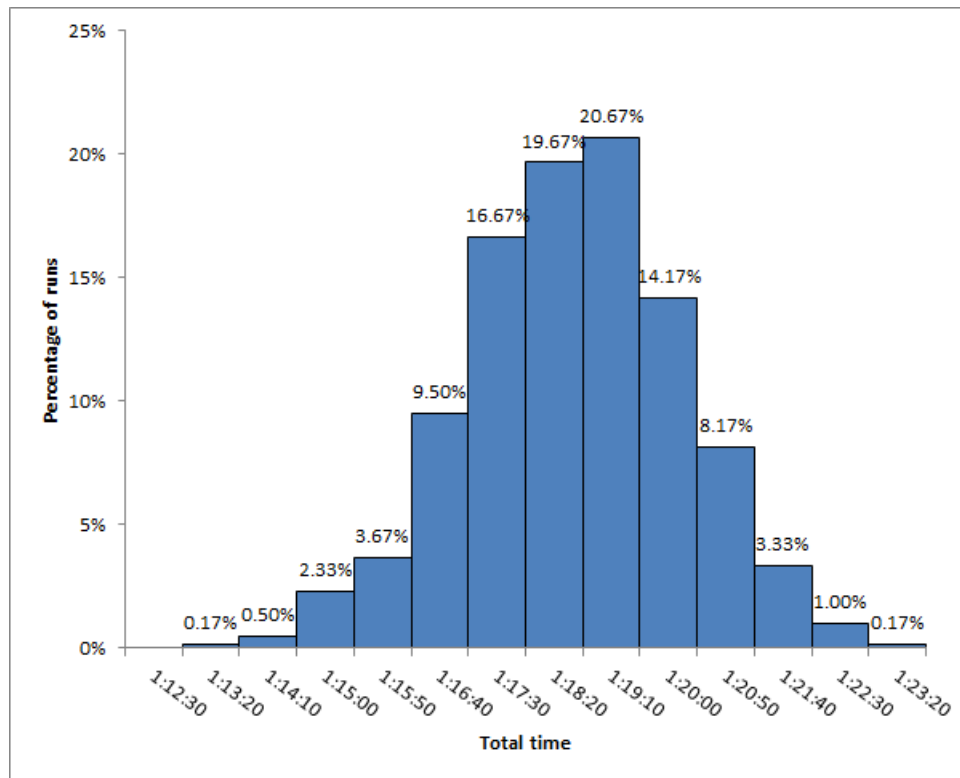


Figure 9.1: Total time distribution for user 1, strategy 1

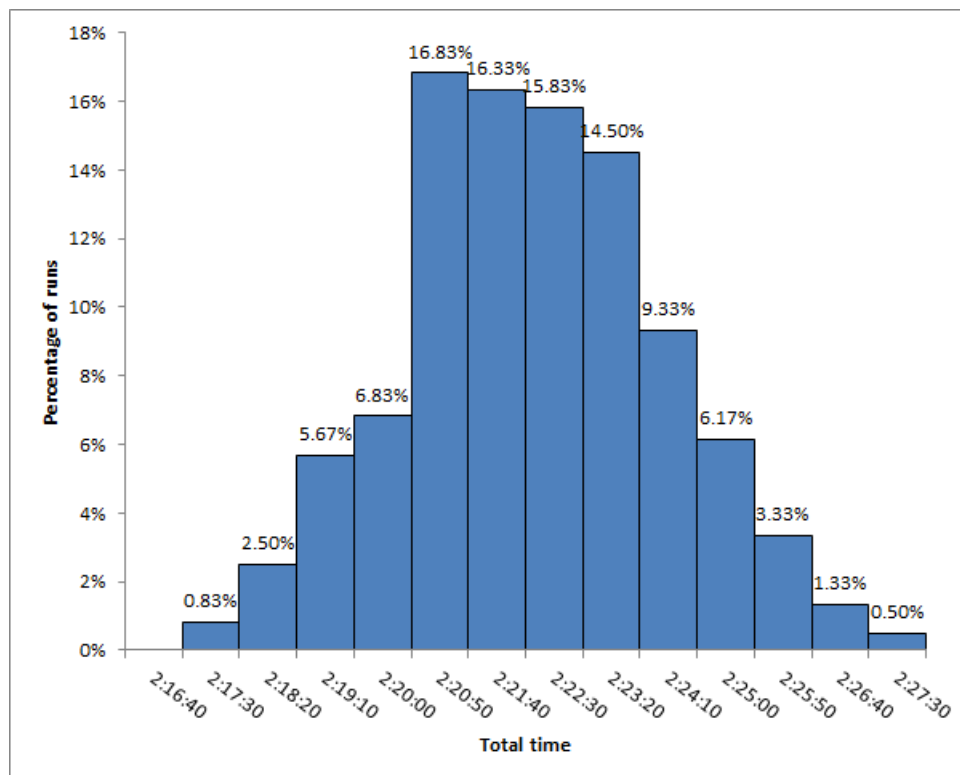


Figure 9.2: Total time distribution for user 2, strategy 2

9.3.2.1 Changing The Strategy for User 1

We start by changing the model for user 1 to simulate her doing strategy 2 (the normal strategy of user 2) for filling in alerts onto the spreadsheet. This involves the changing of the skills of user 1 into the order of the steps for strategy 2, which were presented in the previous subsection. To keep the process closer to the profile of the real experienced user, we still dismiss the time for retrieving questions (unlike for user 2 doing strategy 2, who would retrieve them before looking for the information helping to answer them from the Alerts page). Also, we notice that the purpose for retrieving alerts from the Alerts page is different for the two strategies. In particular, strategy 1 has the purpose of initially remembering all of the alerts, or their absence, and if later any of them, or their absence, is forgotten for answering a question, it has the purpose of remembering only that alert or about its absence. As shown in Table 8.1 from the previous chapter, strategy 1 therefore has two ‘retrieve alerts’ external skills, with different allocated times (durations). Instead, strategy 2 always retrieves only the alert, or its absence, which would help answer a question, and thus there is only one ‘retrieve alerts’ external skill, as was shown in Table 8.2. To make it possible for user 1 to have strategy 2 under these circumstances, we use the time of user 2 for her for the single ‘retrieve alerts’ external skill required by strategy 2.

After changing the model of the user this way, I ran another high number of simulations, until the results from the histograms were stable (according to the convention stated earlier). The obtained histogram is presented in Fig 9.3. According to it, user 1 would need at most 1h 35min if employing strategy 2, which is still much less time than the currently allocated time. The analysis also reveals that to offer her less time, say 1h 30min, would result in her exceeding this time in a high number of work days (i.e. runs), representing 34.16% of their total number. This is very risky, and shows just how important the extra 5 minutes are for the user.

I have mentioned in the statement of the basic case study, and also in the previous chapter for the real context, that other COs use a third strategy (strategy 3) for filling in alerts onto the spreadsheet. This strategy can be summarised as follows: once on the second page of the spreadsheet, the user opens the website tab, goes to the Alerts page, writes down on paper all of the alerts, or about their absence, then returns to the spreadsheet tab and answers the questions with ‘yes’ or ‘no’ according to the notes from the piece of paper. To change the (initial) model for user 1 to simulate her doing strategy 3, we need to change some of the user’s skills and concepts. In particular, the user

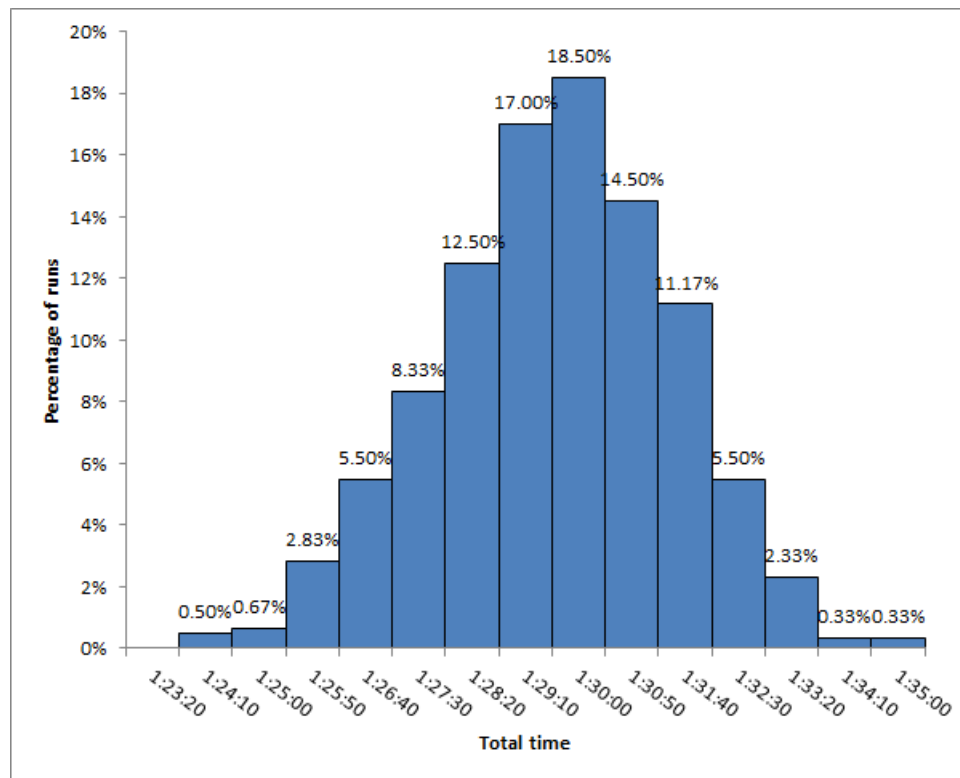


Figure 9.3: Total time distribution for user 1, strategy 2

does not need to set herself the purpose of remembering alerts, or about their absence, when going to the Alerts page (i.e. remove the ‘percepts to remember’ component). She will not need as a subskill, and external skill, the retrieval of alerts from the Alerts page, and instead she will need an external skill for writing down the alerts. I used for the duration of this external skill the mean and standard deviation of the durations of another real CO observed during my study from Lothian, who was using strategy 3. I set up its lower bound to a reasonable time in which a person could write down one alert in shorthand. We also change the implementation for carrying out external skills (from Java) to set that the user does not want to ever forget the written down alerts from the Alerts page, or about their absence (i.e. set the percepts for them as ‘from external’). Together with this change, we also remove the simulated user’s concept of checking whether a question has an answer in Alerts, and her skills such that she will never return to the Alerts page a second time in such a case (as she will never forget the information).

The new stabilised results are presented in the histogram from Fig. 9.4. They show that, if user 1 uses strategy 3, she would need at most 1h 25min 50s, but that offering her 1h 25min would not be very risky, as only in 0.33% of work days she would need

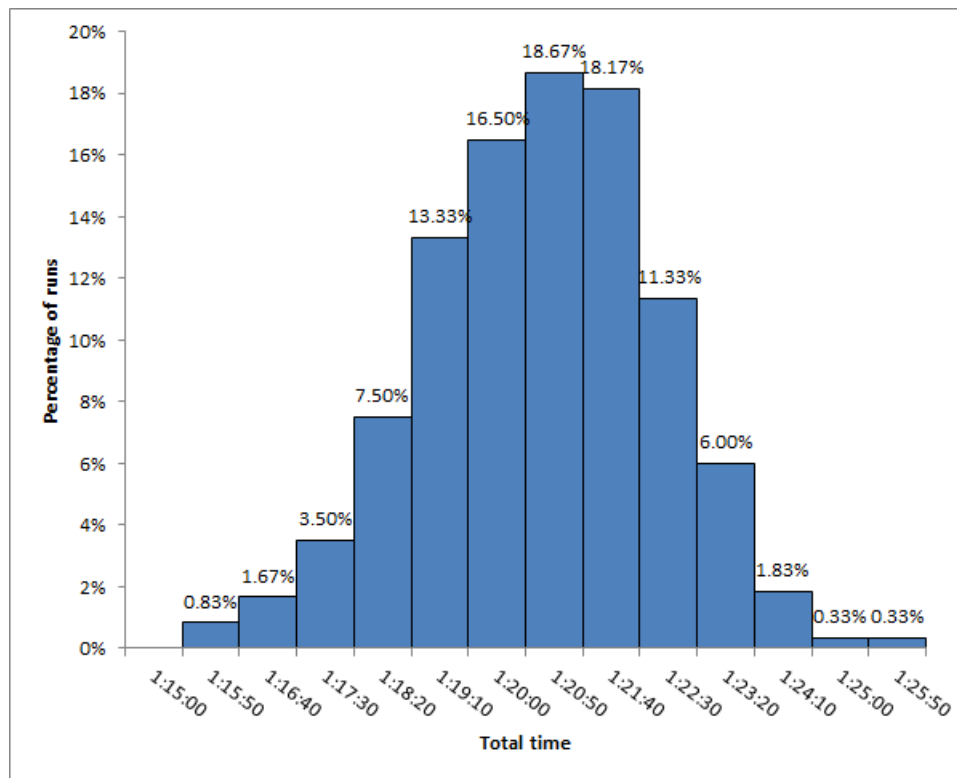


Figure 9.4: Total time distribution for user 1, strategy 3

more than this time. To offer her instead 1h 20min would be too risky, as in 56.66% of work days she would exceed this time.

9.3.2.2 Changing The Strategy for User 2

We will start by changing the model for user 2 such that it can simulate the use of strategy 1, of user 1, for filling in alerts onto the spreadsheet. To do this, we change the skills accordingly. As the real CO having inspired user 2 would always look at a question before looking for the alert corresponding to it from the Alerts page (or other pages), we keep this consistent by adding that the simulated user 2 will retrieve the question in case she forgets the alert, or its absence helping to answer it, for strategy 1. We keep the details (including time) from strategy 2 for the external skill of retrieving the question. As noticed in the previous subsection, the purpose for retrieving alerts from the Alerts page is different for the two strategies. Similarly to how we dealt with this for user 1, for implementing strategy 1 for user 2 we use the times of user 1 for retrieving alerts the first time, and if alerts, or their absence is later forgotten.

Another high enough number of simulations produces the histogram provided in Fig 9.5. The histogram shows that user 2 would need at most 2h 10min 50s, and that

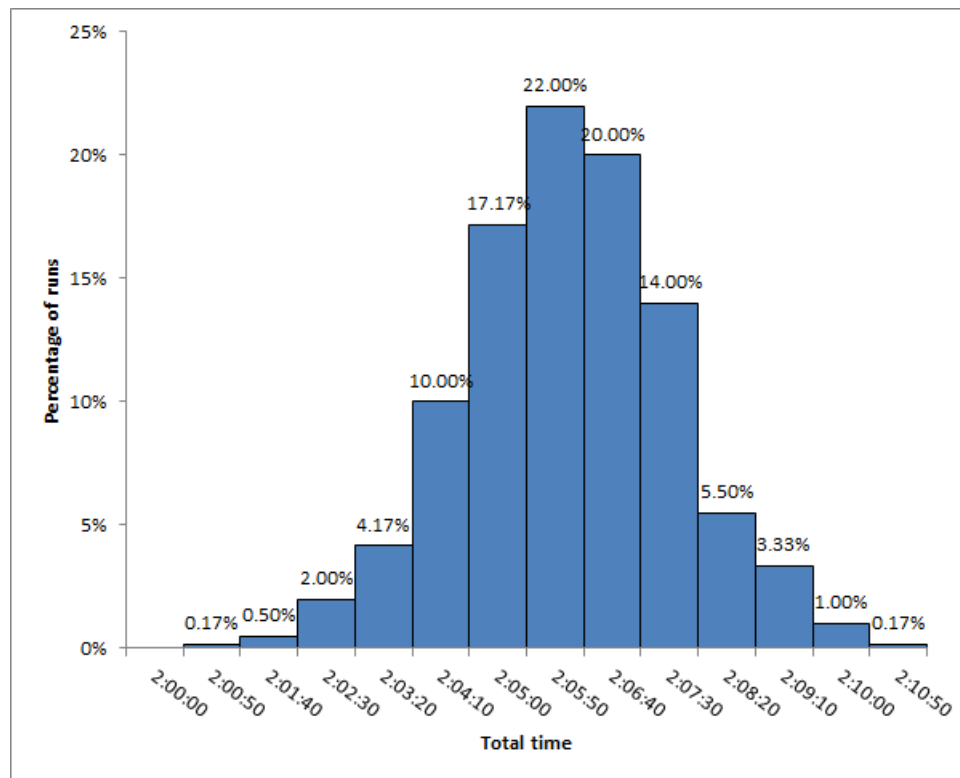


Figure 9.5: Total time distribution for user 2, strategy 1

allocating for her 2h 10min would be enough in almost all of the work days.

We change the model for user 2 to simulate her using strategy 3 by using the same skills and concepts related to this strategy as for user 1. For both of the users the time to write down alerts is taken from a third real user, which makes them equal in this respect. The histogram obtained from running another high enough number of runs is provided in Fig. 9.6. We can see from it that, if using strategy 3, user 2 would need at most 2h 10min. To offer her less time than that, say 2h 5min, would be too risky, as she would exceed this time in a very high percentage of work days, 56.98%, so the extra 5 minutes are essential for the user.

9.3.3 Addressing Question 3- Comparing the Results Between Users for Each Strategy

To understand what time users with different levels of experience would need, if they use each of the strategies, we can compare the results of the two modelled users per strategy.

If we compare the users' results for strategy 1, we find that the modal time for user

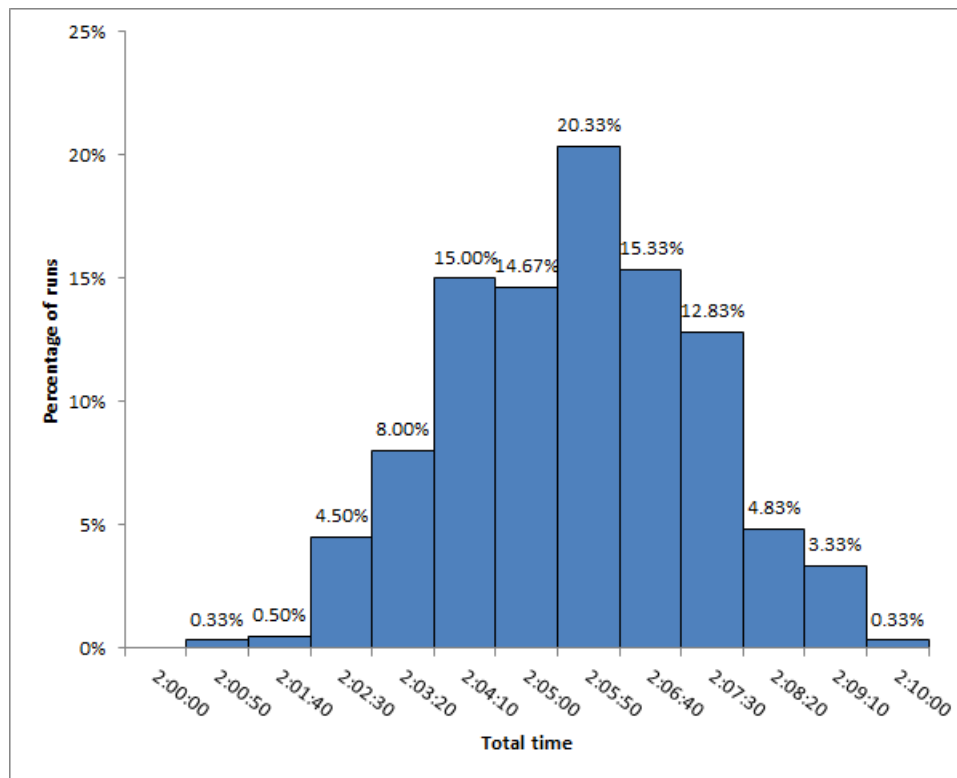


Figure 9.6: Total time distribution for user 2, strategy 3

2 is 46min 40s higher, and that her median time is 47min 30s higher, than those for user 1. With regard to the optimal time to have assigned, the safest solution would be to offer 50 more minutes to user 2. As we have used the same time for retrieving alerts for the two users, these differences are given by the longer time required by user 2 for the other steps (from device actions and external skills). Being slower, user 2 also forgets information more frequently than user 1. When using this strategy, she will forget the alerts, or about their absence, which would answer a question more frequently, and thus need to return more frequently to the Alerts page.

For similar reasons, with regard to strategy 2, user 2 is slower with a difference of 50min 50s in terms of modes, and of 52min 30s in terms of medians, as compared to user 1. She would need to have assigned up to 55 more minutes. If we watch the logged results for the simulation runs, we observe that user 2 moves backwards and forwards between the website and the spreadsheet tab to fill in the alerts according to this strategy, and never, or very seldom, remembers an additional alert, or about its absence, than the one that she had set out to remember from the Alerts page. This is because she usually spends more time in filling in the previous question than that which would allow her to remember this information. Instead, user 1, due to being quicker, will sometimes

remember a second alert, or about its absence, and be able to reply to a new question once having replied to a previous one.

When comparing the results for strategy 3, we find that user 2 is slower with a difference of 45min in terms of modes and medians, than user 1, and that she would need to have assigned 40 more minutes. The difference is smaller here as compared to the other strategies, because, as they are writing information on paper and using it from there, the users will not rely on their memory, which causes the larger differences from the previous points.

9.3.4 Addressing Question 4- Comparing the Results for Each User

A comparison of the time distribution for the three strategies for user 1 is provided in Fig. 9.7. It clearly shows that the strategy which would result in the best system and work process efficiency for user 1 is strategy 1, her own strategy, followed closely by strategy 3 (the median of which is 3.19% higher, and the mode of which is 2.10% higher than those of strategy 1), and followed by strategy 2 (the median of which is 14.89% higher, and the mode of which is 13.68% higher than those of strategy 1). For carrying out strategy 3, the management could offer her with little risk the same total time as for strategy 1. Strategy 2 instead requires an allocation of 10 more minutes to the user, and would result in an increase of her median time with 11min 40s, and of her modal time with 10min 50s. For all of the strategies user 1 would manage her work well with the 2h 30min that she has currently allocated, and it is uneconomical for the healthcare organisation to offer her this much time.

For user 2, the comparison of time distributions for the three strategies (Fig. 9.8) shows that strategy 3 would very marginally result in the best system and work process efficiency, followed very closely by strategy 1. Although having the same median and modal times, strategy 3 has a slightly better chance for lower total times as compared to strategy 1. Strategy 2 is the worst option for the user. As compared to it, the median time of strategies 1 and 3 is lower by 16min 40s (11.69%), and the modal time by 15min (10.65%). As we have seen in subsection 9.3.2.2, the management could allocate to user 2 for both strategies 1 and 3 a similar time, as there is an extremely small risk to offer her 2h 10min for strategy 1. Using her own strategy the user would barely manage to monitor the 20 patients within the 2h 30min, while the other strategies make this well achievable, and this time could even be reduced by 20 minutes.

The longer time that user 2 spends in doing her actions on the system results in a

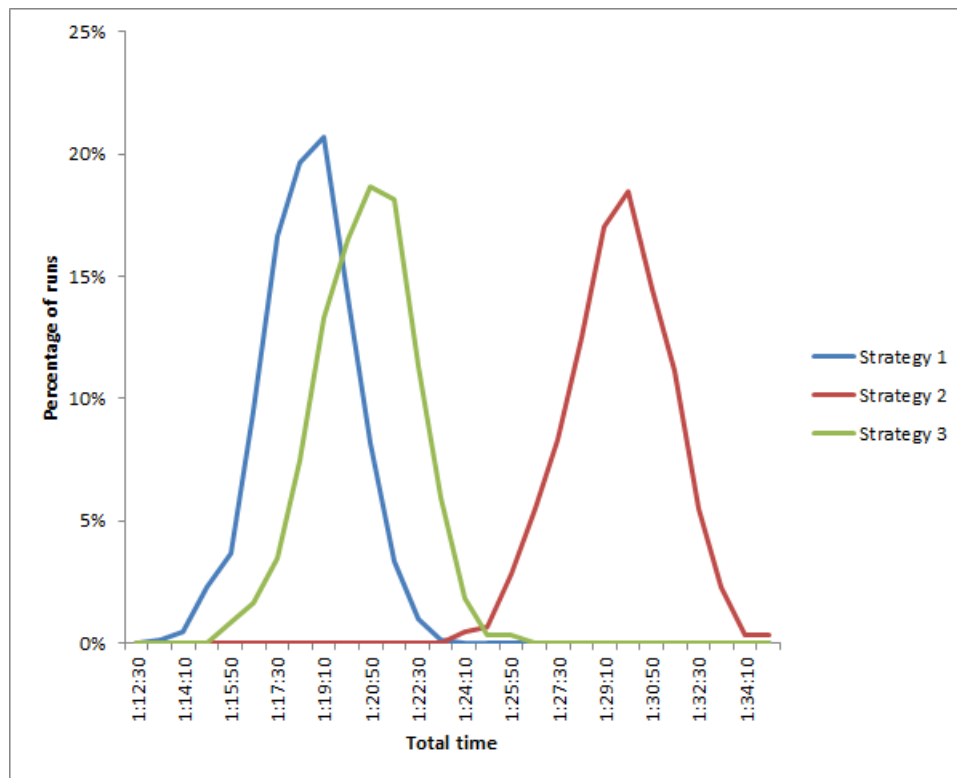


Figure 9.7: Differences between the time distributions for the three strategies for user 1

greater reliance on her memory, as compared to user 1. Therefore, the fact that strategy 3 is marginally the best for this user is not surprising, given its minimal requirements on her memory (she writes the information down instead). User 1 spends slightly less time using strategy 1, because she is generally quicker and relies less on her memory. Her quick actions for looking at the alerts (when filling them in or entering the outcome on the spreadsheet) compensate the time which would be saved by writing them down on paper in her case.

9.3.5 Addressing Question 5- Comparing All of the Results

Strategy 1 was clearly the best for user 1, while both strategies 1 and 3 were similarly good for user 2, with strategy 3 very marginally the best. As beginner users would also gain experience after some time with the system, this could motivate the inclusion in training sessions of advice for using strategy 1. Nevertheless, other aspects such as the chance for making mistakes, patient safety and confidentiality should also be considered.

Strategy 1 seems to be a good idea, especially for experienced users, but only if they have good memory. Importantly, it may lead to mistakes if users rely too much on

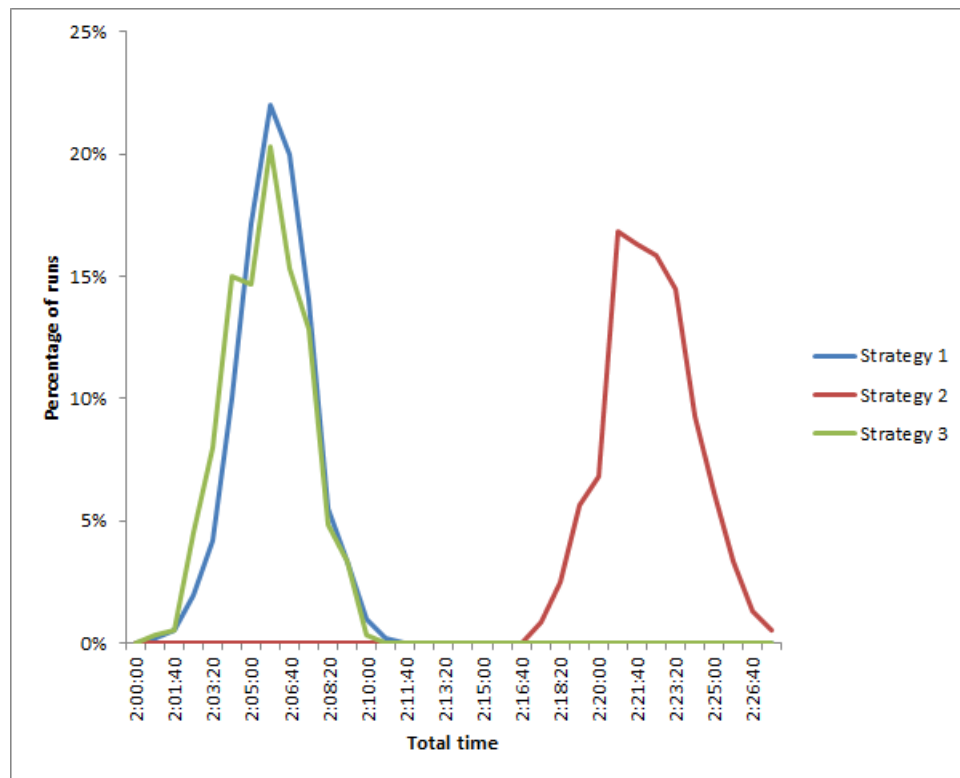


Figure 9.8: Differences between the time distributions for the three strategies for user 2

their memory, especially if they work in a busy environment or are interrupted. Such mistakes could result in incorrect decisions being made about a patient's state, this having implications on patient safety. It also requires a certain level of confidence in one's own memory for employing it. Users who are more cautious, and therefore go backwards and forwards to double check the alerts, would reduce its efficiency. As a general note, a professional, safety critical system such as a telehealthcare system should minimise user effort (here, including memory) in using it for their work, and therefore a change in the system design could make a great difference on efficiency.

Strategy 3 seems to be beneficial for beginner users, who are slower, and often also less confident in using the system. It does imply though a confidentiality concern: as patient details would be written on paper, a protocol of shredding the paper at the end of the work day would need to be put in place. As a general note, a good system should help its users in managing their workload without the need to use additional tools, especially paper. Therefore, this strategy makes it clear that the system design could be enhanced for improving efficiency.

Out of the three strategies, strategy 2, the least efficient, seems nevertheless to be the safest (as is commonly the case), as the user fills in one question at a time, with a

small chance for mistakes. Also, it does not rely on any external tools. Nevertheless, user 2 using this strategy was very close to exceeding her maximum work time of 2h 30min.

9.4 Second Scenario- Effects of Changes to the Environment

In this scenario, I will exemplify how management could use my approach for tackling problem 3 from the overview of this chapter.

Let us suppose that, as an effect of the investigation from the previous scenario, the management have understood that strategy 2 is the safest. Unfortunately, it is also the slowest. Therefore, they would like to investigate ways to improve the efficiency of the system and work process for it. A major criticism of the telehealthcare system is the fact that users need to move in between applications, especially the website and the spreadsheet, for doing their work, and some users need to minimise their windows or write information down on paper for making this easier (as we saw for the real system in Chapter 3). The management are therefore considering, as the first potential solution, purchasing large wide screen monitors for the users, which they could use to look at two tabs at once. They would like to investigate whether this solution would affect the efficiency of the system and work process enough, both for experienced users and beginners, to motivate investment in it.

A wider screen would eliminate the need for users to move in between tabs which, as we have seen from the previous scenario, can be frequent, and incurs durations which reduce how long they remember information. Moreover, if the screen is placed ergonomically, users would not even need to move their heads for looking at the different tabs.

To analyse this problem, we take the two user models for user 1- strategy 2 and user 2- strategy 2 in turn, and change them to reflect how they would work differently given the new monitor. In particular, their skills should consider, instead of the movement between tabs, their looking at the different tabs as a subskill ¹. I have set for it a mean duration of 180 ms, according to the findings on eye movement times between parts of the screen from a study ([63]), a reasonable 200 ms standard deviation and 100 ms

¹I am making the assumption that they have the tabs as already set up when they start their monitoring work on the system

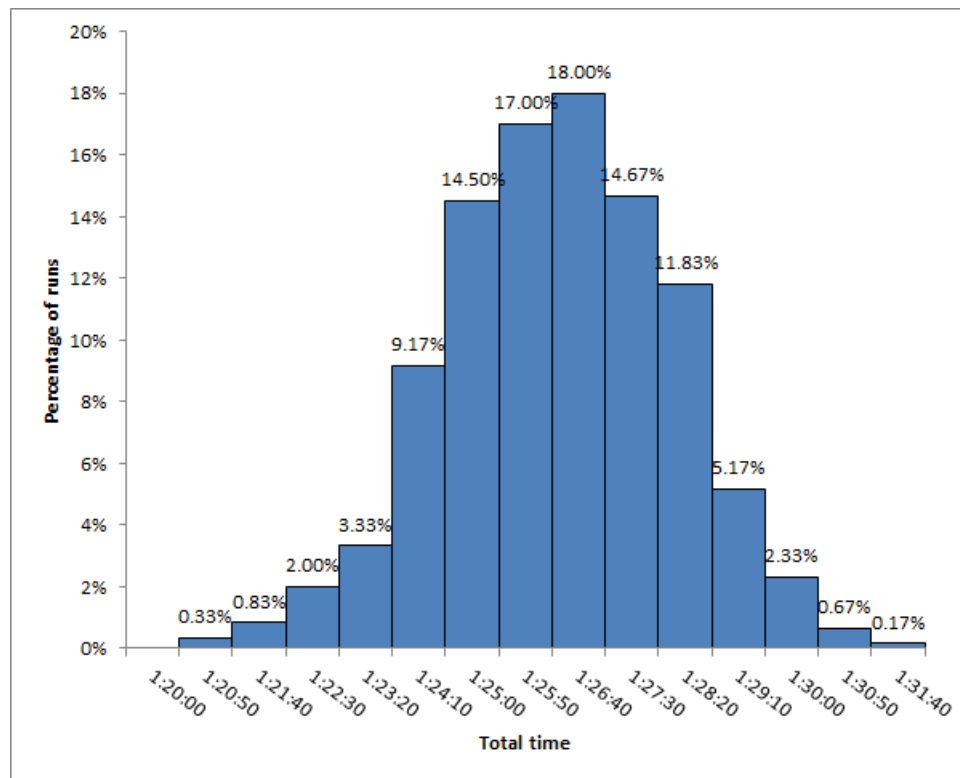


Figure 9.9: Total time distribution for user 1, strategy 2, large screen

lower bound. With regard to the device, we must change it to display the different tabs to the user when requested.

The stabilised results of running a high number of simulations of user 1 using strategy 2 and a large monitor is provided in Fig. 9.9. The histogram shows that the user would need at most 1h 31min 40s for her work, but that the management could decide to offer her 1h 30min because she would only exceed this time in 0.84% of work days. As compared with her initial time, when using the standard screen (Fig. 9.10), she could therefore be offered 5 less minutes in this scenario. Although the difference between her median times and modal times is small- 3min 20s, it allows for slightly lower time allocation, which would be more economical.

The results of running a high number of simulations for user 2 with strategy 2 in this scenario (Fig. 9.11) show that she would need at most 2h 17min 30s to handle her work, so the management would ideally offer her 2h 20min. Nevertheless, she would exceed 2h 15min only in 7.66% of work days, so it might be considered safe enough to offer her this time instead. As compared to using the standard screen (Fig. 9.12), she could therefore be offered 10 minutes less. The median time for this scenario for her is 10 min (7%) less than the one in which she was not using a large screen, and the modal

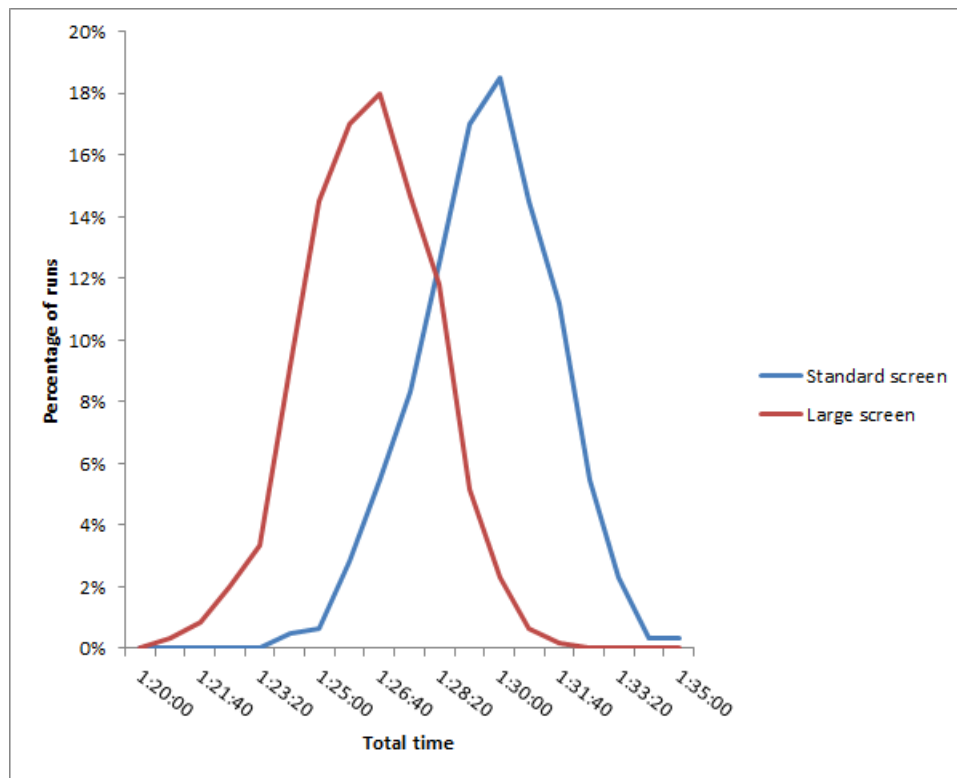


Figure 9.10: Differences between the time distributions for user 1, strategy 2 for standard and large screen

time is 7min 30s (5.32%) less.

Although not affecting the users' average total times greatly, the results show that the use of a larger screen would result in the possibility to allocate 5 minutes less to experienced users, with very little risk, and 10 minutes less to beginners with no risk incurred. These differences are small, but depending on work frequency and the ratio of beginner to experienced users, the cost of new monitors might well pay off. Moreover, monitors are a long term investment, and a call centre would probably find it beneficial to use larger ones also for other purposes than the use of this telehealthcare system. In the worst case scenario in which a decision is made not to invest in larger monitors, the analysis would at least inform about their benefits for the time when new equipment must be purchased.

9.5 Third Scenario- Effects of Changes to the System Design

This scenario addresses problem 4 from the overview of this chapter.

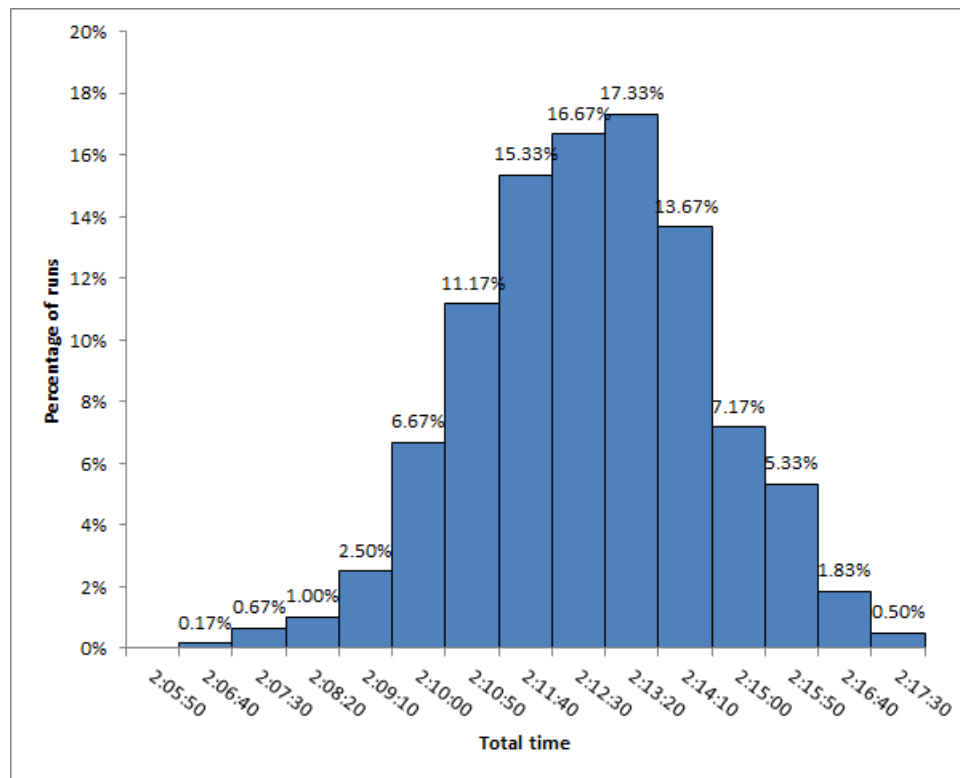


Figure 9.11: Total time distribution for user 2, strategy 2, large screen

As a second potential solution for improving the efficiency of the system and work process for scenario 2, let us assume that the management are investigating whether it would be worth requesting the company which produced the system to make some small improvements to parts of the system design. The management hope that the cost incurred by such a change would be counterbalanced by resulting savings due to a serious reduction of the time which is required by the COs in their work. They would like to investigate whether this is the case. Let us assume that, as starting from problems which have been reported by the COs, they are considering requesting the company to populate all of the alerts for the patient onto the Alerts page. As I mentioned in the basic case study in Section 9.2, some of the alerts are currently placed on other pages for the patient, and the COs need to access these pages to retrieve them. This was a real problem observed by me, and some COs made comments about it, for the real system described in Chapter 3. The management (of the hypothetical deployment) are aware that their proposed change of having all of the alerts populated onto the Alerts page, would lead to a requirement for the COs to close more alerts than at the time being. They are therefore also considering requesting the company a second change, that of adding a functionality for closing down all of the alerts when closing one. As I have

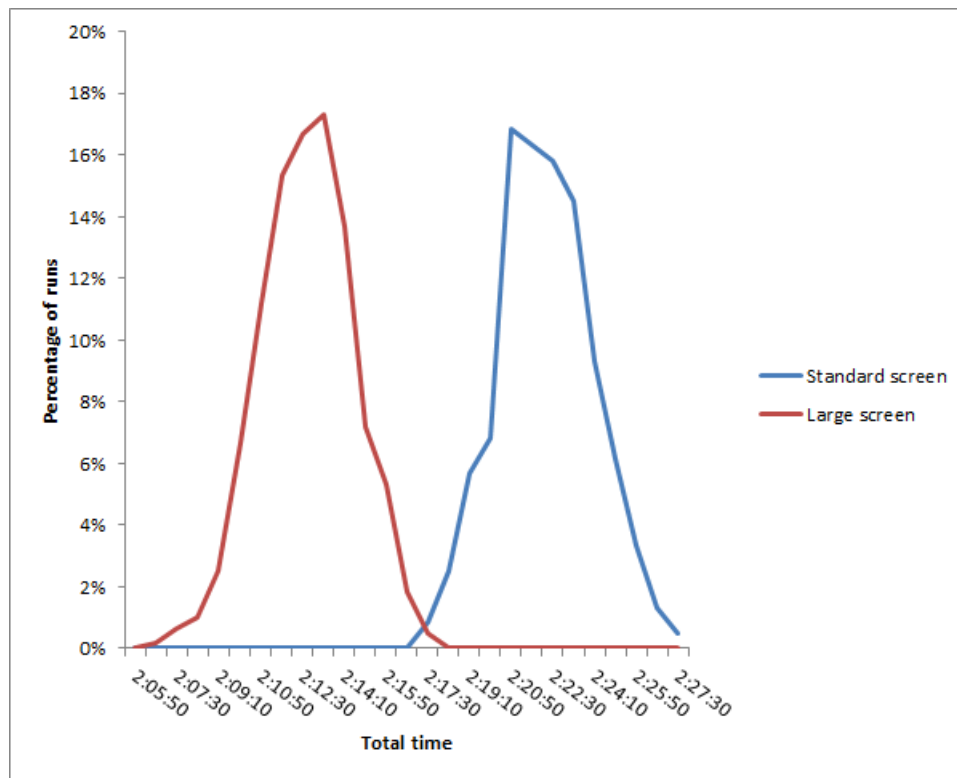


Figure 9.12: Differences between the time distributions for user 2, strategy 2 for standard and large screen

mentioned in Chapter 3 for the real system, some COs and many care providers have explained that having multiple separate alerts on the Alerts page requires them to close them all in turn, although they would enter the same comments and outcome for each.

In my initial models, as following the design of the real system, there are some questions in the spreadsheet the answering of which would require the simulated users to go to the patient's Patient Monitoring, Surveys or Badge pages. The simulated users know where to find alerts for each type of question. To represent the management's proposed changes to the system design, we change the device model such that, when the device action of clicking a button for reaching the Alerts page takes place, the device state will become populated with all of the patient's alerts. We consider as an exception the alert from the badge, which involves the user looking into a list of notes about the patient, and thus cannot be easily imported into the table of alerts. We also replace the button which currently saves and closes down a selected edited alert from the Alerts page with one for saving and closing them down all at once, and add the associated device action and functionality for changing the device state in this case. I considered that the device action would take the same duration as for saving one single alert. For

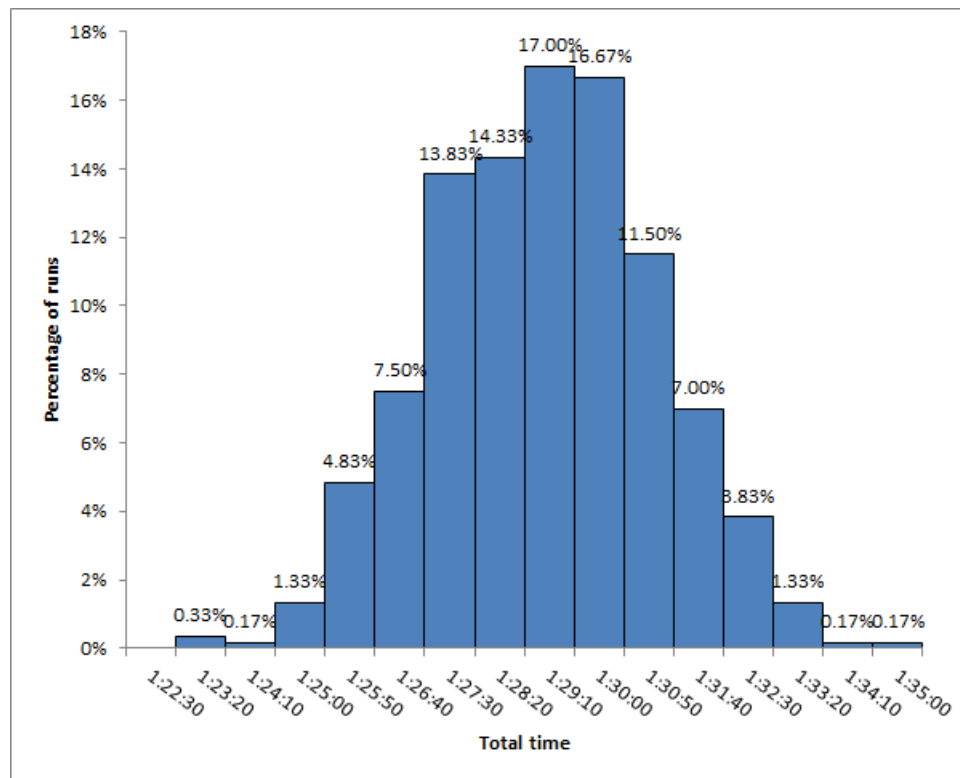


Figure 9.13: Total time distribution for user 1, strategy 2, all alerts on one page

the user models, we remove all of the skills and concepts which have to do with the movement and retrieval of information from the Patient Monitoring and Surveys pages, as the users would no longer need the knowledge to reach them.

Changing the device model, and the model of user 1 using strategy 2 this way, we run another high enough number of simulations and obtain the stabilised results from Fig. 9.13. The histogram shows that user 1 would need up to 1h 35min, and that it would be risky to offer her less time, as she would exceed 1h 30min in 24.53% of work days. She would thus require the same allocated time as for the initial design. When comparing the two designs for her (Fig. 9.14), we notice that the median times and the modal times are very close, with a negligible 50 s improvement for the new design, which has a slightly better chance for shorter total times.

Stabilised results obtained from a high enough number of simulation runs with user 2, scenario 2 (Fig. 9.15) show that, if the system design is changed as proposed, the user would require at most 2h 26min 40s, but 2h 25min would be a perfectly viable time allocation. It would be almost risk-free to offer her this time, but it would still be the same time as for the initial design. As opposed to the results for user 1, the comparison of the two designs for user 2 (Fig. 9.16) shows a better, but still not considerable, time

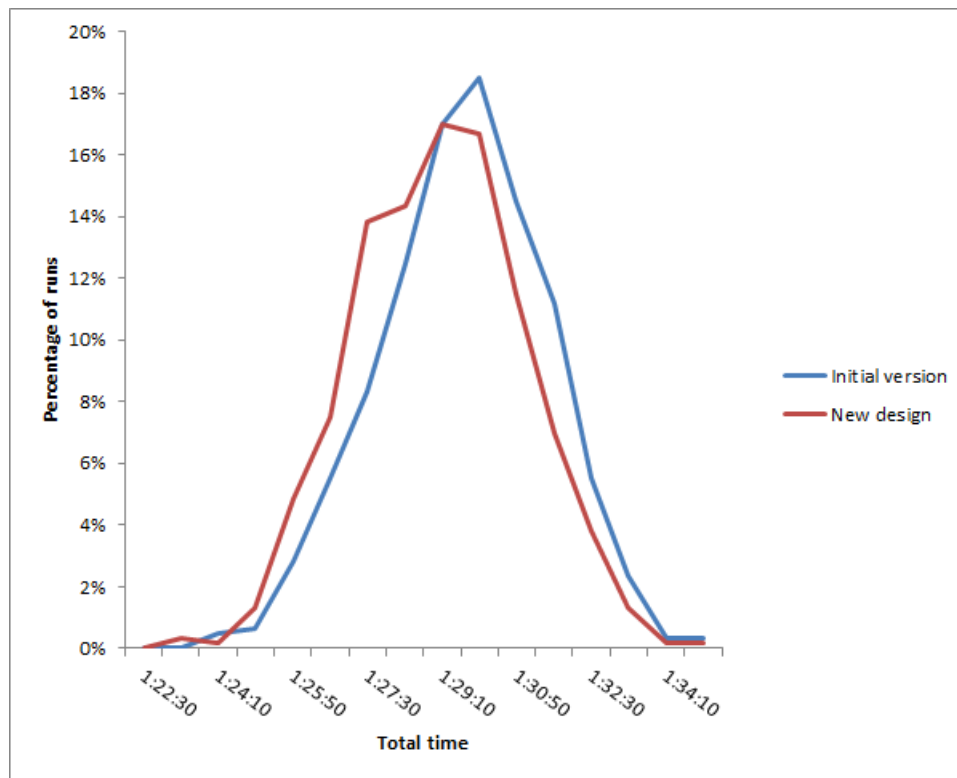


Figure 9.14: Differences between the time distributions for user 1, strategy 2 for initial version and new design

improvement. In particular, while the modal times for the new and old design for her are the same, the median time for the new design is improved by 2min 30s, representing only a 1.75% improvement.

In conclusion, the changes to the system design proposed by the management, although more comfortable to the users (which might be an important consideration), would only lead to a very minor improvement in the efficiency of the system and work process. This improvement is slightly better for beginners, but it levels out as they become more experienced. It would therefore probably not be worth investing in making these small changes to the system design, considering the potentially high costs. Instead, the management could consider a higher investment which would completely eliminate the need for the COs to fill in alerts onto the spreadsheet: that of the spreadsheet being automatically populated by alerts from the website through integration. As we have seen in Chapter 3 for the real system, all of the COs and their team leaders saw integration between the website and spreadsheet as desirable.

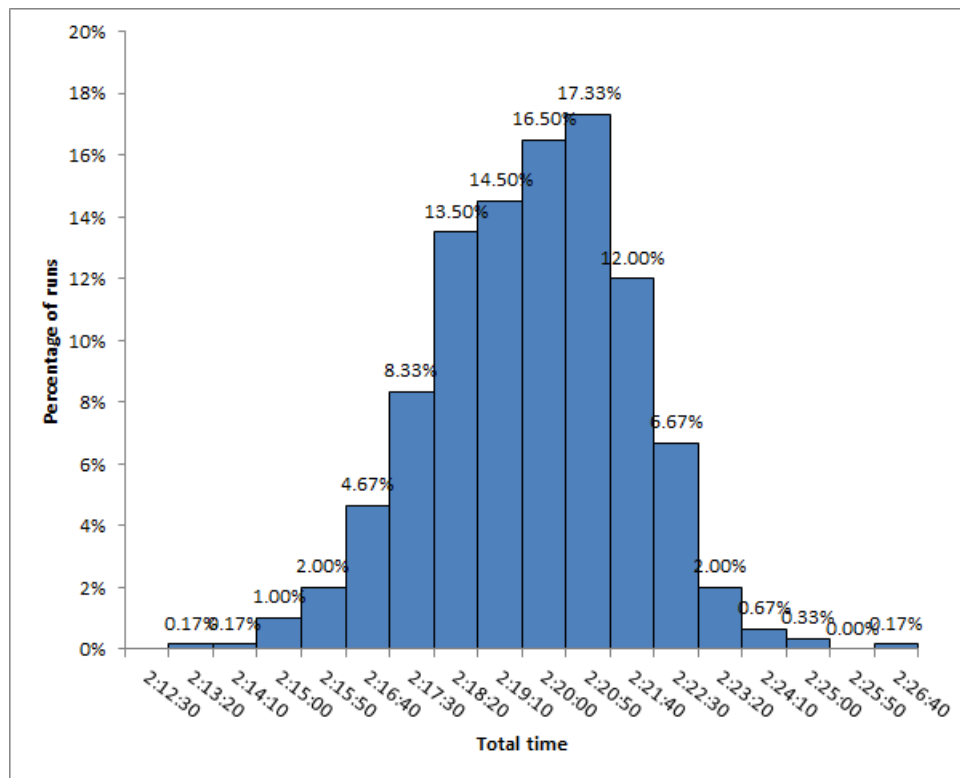


Figure 9.15: Total time distribution for user 2, strategy 2, all alerts on one page

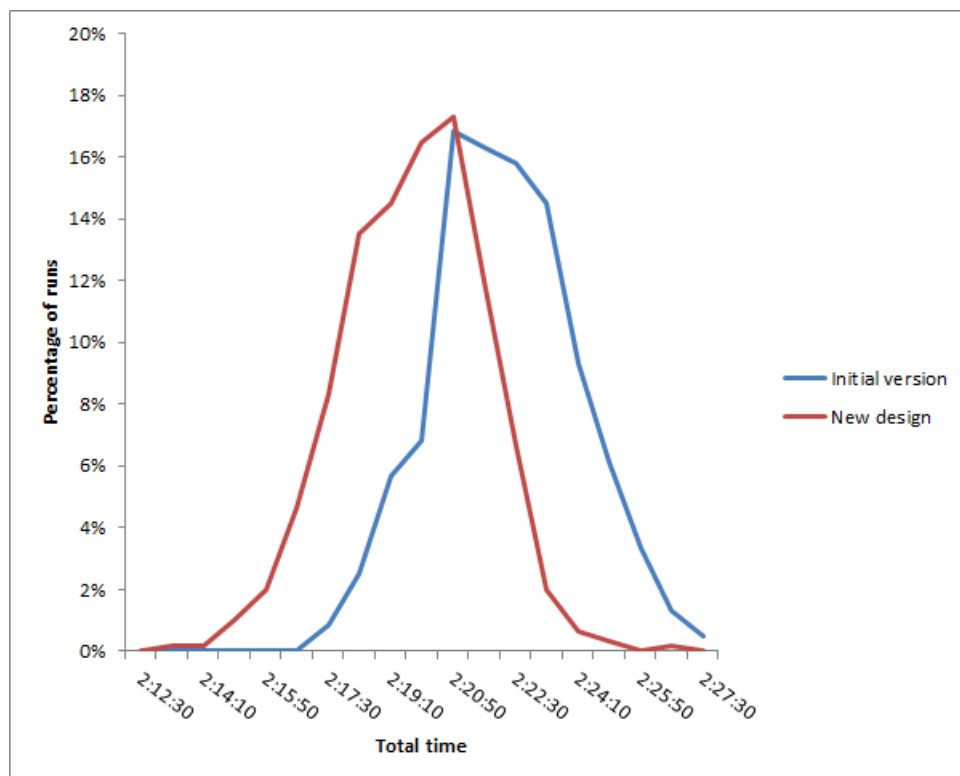


Figure 9.16: Differences between the time distributions for user 2, strategy 2 for initial version and new design

9.6 Fourth Scenario- Effects of Foreseeable Workload Changes

This scenario addresses the last general problem described in the overview of this chapter.

Let us suppose that, as an effect to the analysis which was described within the previous two scenarios, the management decide not to invest in any changes to the work environment or system design for the time being. As winter is coming, and with it a good chance for epidemics which would increase exacerbations for both COPD and CHF patients, they would like instead to focus their attention on mitigating risks. In particular, they would like to analyse the way that beginner and experienced users who use strategy 2 would be affected by a considerably increased workload, in terms of more frequently alerting patents, in the scenario of a bad epidemic.

The increase in workload cannot be characterised by a single figure. As the epidemic would influence patients differently, their probability of having each possible combination of alerts will not increase uniformly. Given this complexity, the effects of workload changes on system and work process efficiency are not foreseeable by the naked eye. My approach can help in this respect, because it allows a modeller to repeatedly, and easily, change the characterisation of the workload from the inputs, and run simulations with each to understand how they would affect efficiency.

Let us suppose that the management would like to explore the effects on efficiency of one possible change in workload: the number of alerting COPD patients increasing from 17 to 28, thus to 93.33% of the total number of COPD patients (30), and the number of alerting CHF patients increasing from 3 to 9, thus to 90% of the total number of CHF patients (10). They also define the number of patients highlighted in each colour, as given by their combination of alerts: 24 red and 4 orange for COPD, representing 80% and 13.33% of the total 30 COPD patients, and 8 red and 1 orange for CHF, representing 80% and 10% of the total 10 CHF patients. Moreover, they consider that the number of patients highlighted in each colour having each of the possible combinations of alerts increases in a different way, as represented in Table 9.1, where the initial values are the ones from the initial list of settings from Appendix G.

To explore the effects on efficiency of this changed workload, we only need to make the above changes to the list of configurations provided as inputs to the run of the models. We then simulate users 1 and 2, using strategy 2, in handling this new workload in their monitoring work.

Condition	Alert combination	Highlight colour	Initial value	Value for epidemic
COPD	1	red	4	8
COPD	2	red	3	10
COPD	3	red	1	6
COPD	4	orange	3	1
COPD	5	orange	4	2
COPD	6	orange	2	1
CHF	1	red	1	5
CHF	2	red	1	3
CHF	3	orange	1	1

Table 9.1: Increase in the number of patients highlighted in each colour, having each combination of alerts

The histogram obtained after performing a high enough number of runs for user 1- strategy 2 is provided in Fig. 9.17. It shows that user 1 would need at most 3h 21min 40s, so 3h 25min would be the ideal time to allocate to her. Nevertheless, she could have allocated, with infinitesimal risk, 3h 20min for handling her workload in this case, and there is a greater risk for her to exceed the total time in 13.83% of work days if management offers her 3h 15min instead. In comparison with the management of her current workload, her median time would increase by 1h 42min 30s, and her modal time by 1h 43min 20s.

According to the histogram obtained after another high number of runs for user 2- strategy 2 (Fig. 9.18), user 2 would need at most 5h 6min 40s for her work, but the management could easily allocate to her 5h 5min, as she would not exceed this time in almost any work days. It would probably be too risky to offer her 5h, as she would exceed this time in 21.5% of work days. She would therefore need to have allocated at most 1h 45min more than user 1. Her median time would increase by 2h 35min 50s, and her modal time by 2h 36min 40s, as opposed to her management of the current workload.

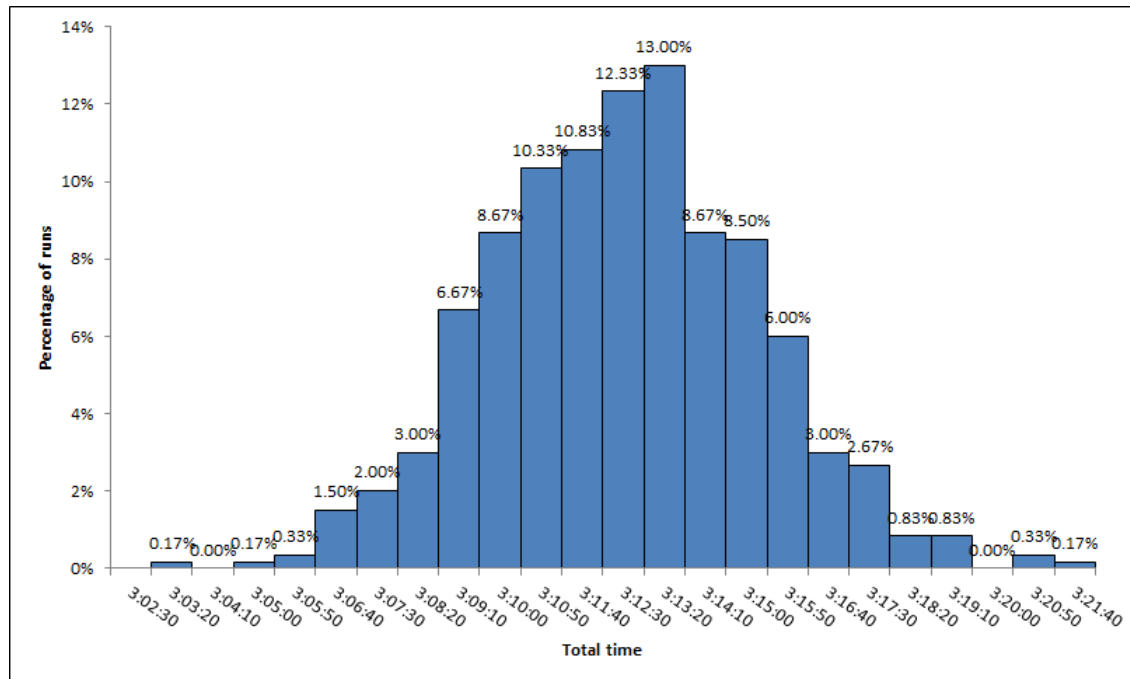


Figure 9.17: Total time distribution for user 1, strategy 2, in the case of an epidemic

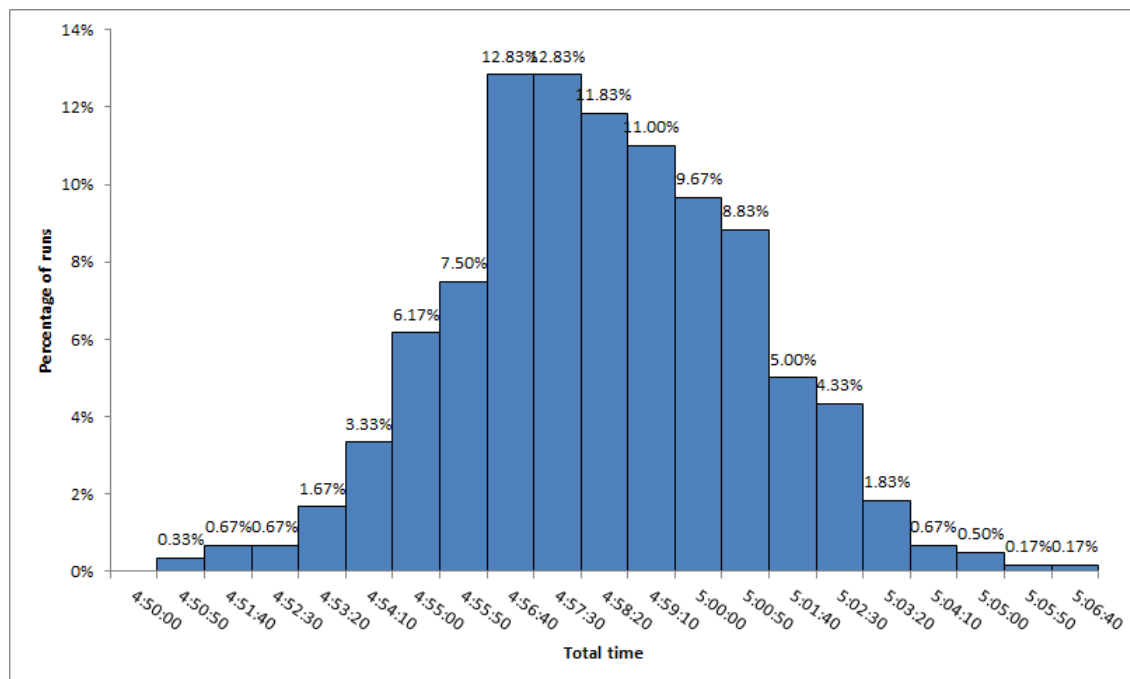


Figure 9.18: Total time distribution for user 2, strategy 2, in the case of an epidemic

9.7 Conclusions and Other Uses

In this chapter, I have illustrated how my methodology and modelling approach could be used within a call centre deployment to help managers address a series of problems related to the current and potential efficiency of a telehealthcare system and work process surrounding it, given differences in user experience and ways of working, proposed changes to the environment or system design, and foreseeable changes to the workload. Together with considerations of costs, risks on patient safety, CO responsibilities and work organisation, my approach can be a useful tool for deciding on time and resource allocation, the feasibility of investment in changes, and for mitigating risks.

The examples highlighted only one scenario in which my work could be used, for management in an existing call centre deployment. The following are other scenarios for which I believe it could constitute a real asset:

- Investors analysing whether it would be worth investing in deploying a telehealthcare system within a potential site. In case information about how users used a similar system there in the past could be collected, they could use my approach to analyse whether work would be manageable, given reasonable resource expenditure, and, if not, what would need to be changed. This is particularly relevant for projects which invest large sums in scaling up telehealthcare, for which there is a high risk of failure in heterogeneous deployment sites.
- Managers preparing a deployment in which the telehealthcare system will be rolled out, such as to ensure high system and work process efficiency there.
- Software developers evaluating a telehealthcare system during summative usability evaluation. By only using data from a few representative test users, they could use my approach to discover inefficient areas of the system, and be able to cheaply investigate what would be useful design improvements. By using the logs resulted from simulations, they could gain a better understanding of likely user behaviour, and thus be able to build the user's mental model, without the need for further usability testing. They could compare this model with the system model to propose design improvements. Findings from using my approach could also be useful for writing the help and documentation of the system. While my approach would not be as good as usability testing, it could provide an acceptable approximation of user behaviour, at lower costs. Overall, using my approach

could help ensure a better fit with user needs and requirements once the system is rolled out.

Of course, my modelling approach could also be used for automating usability evaluation within earlier phases of design, which are not the subject of this thesis. Nevertheless, as quick design decisions are often required then, developers could be better off using quicker and simpler usability evaluation approaches, such as inspection methods, or a high level and informal version of GOMS ([62]). As explained in Chapter 2 section 2.5.2.2.4, cognitive architectures like the one used for building my user models can offer more flexibility in what can be modelled, and more accurate and detailed results, which is why it may be worth investing in using them for post-development evaluation purposes.

Chapter 10

Discussion, Conclusions and Future Work

10.1 Overview

In this chapter, I will first discuss some issues which should be considered for commercialising the work presented in this thesis. Some of these issues have been resolved during my development of the examples presented in the previous chapter, while others require more research and a collaboration with specialists, and are proposed as future work. I will then draw conclusions, and summarise the contributions of the thesis. Finally, I will make some recommendations for policy makers for making Scotland a friendlier environment for carrying our health IT research.

10.2 Issues in Using the Methodology and Modelling Approach Commercially

The work presented in this thesis is novel from the perspective of applying modelling and simulation techniques for predicting efficiency within a telehealthcare context. My vision is that, if commercialised, it could become a very useful tool for healthcare organisations and software developers, due all of the reasons presented in the previous chapter. However, if this work is to be commercialised, a series of issues need to be addressed.

10.2.1 Issues Related to Data Collection

I will not include in this section general issues involved in any empirical study carried out in NHS premises, such as obtaining permission from the NHS for the study and anonymising the data. Instead, I will focus on specific problems which should be considered for using my approach, which have also been briefly presented for the methodology in Chapter 4:

- **Finding a compromise between too little and too much data for the models:**

Insufficient data is a limiting factor in obtaining accurate efficiency predictions, and therefore the users of my modelling approach should ideally have the possibility of collecting data from large enough real reference and potential deployment sites. Just how large these sites would need to be, thus the quantity of data of each input type needed, could be determined by performing a sensitivity analysis [59]. In particular, we could perturb the inputs of the model synthetically, one at a time, to analyse the effects on the efficiency predictions. This could help us identify those types of inputs for which small changes would lead to significant variability in the efficiency predictions, and thus for which we would need to collect more data to reduce uncertainty. Using such an approach and then collecting all of the needed data for each type of input would be ideal, but nevertheless not always feasible, or even desirable, due to several reasons. First of all, data collection can be a lengthy and costly process, especially since it should ideally involve a complex study methodology (as I will explain below). Secondly, depending on the needed reliability of the results, as given by the risks of the potential deployment which is investigated, more or less data may be required. For example, in a low probability, high risk scenario such as an unusually severe epidemic, obtaining the right amount of data for being able to accurately predict efficiency for mitigating risks would be critical. Less data may be required for understanding how efficiency is influenced by different ways of working for training purposes. Training should only suggest, but not impose, optimal work strategies, and users would ultimately use the strategy that is best for them. Lastly, but very importantly, it may not even be possible to collect the ideal amount of data for the purposes of the study, especially in the case of small scale deployments during telehealthcare pilots. In my case, the small scale of the deployment meant that, although I observed and interviewed the great majority of COs, the data was still insufficient and I needed to sometimes make assumptions.

In particular, no two COs worked the same way.

- **Investing in a complex study methodology, involving a combination of data collection methods:** Data on the order and timing of the tasks performed by the users on the system (to be used for device actions) can be best collected through *interaction logging*, either by having the system collect such data itself, or the user's browser being instrumented with logging facilities. With regard to the tasks performed in the environment, the same information can be collected (for skills) through *video-recorded observations of users working in the environment*. In the case of my study, I could not use any of these data collection methods due to NHS restrictions regarding software installations, and the limitations of my observer contract (as already mentioned in Chapters 3 and 8). However, in a commercial context I expect them to be possible, with appropriate anonymisation. I also found that, for understanding all of the reasons behind user activities, it is useful to make qualitative observations about organisational culture and values, and therefore an *ethnographic study* could be a good approach. In my case, although my focus was on using observations and interviews to collect data from the call centre, the time that I spent there added considerably to my experience.
- **Establishing a good collaboration with the organisation, in which they agree to share background data and statistics:** The data required for defining work-load characteristics in the list of configurations provided as input for running the models is best collected from the organisation's statistics on monthly work. These statistics may be generated automatically by the system. They may include, for example, the number of patients highlighted in each colour and having each possible type of alert, or having generated each possible outcome during the user's work. For being able to validate the models built for a reference deployment site by following the methodology, data on efficiency can be collected from statistics on hours worked and, for example, the duration of recorded telephone calls. Also, the organisation can provide useful information about the experience of the users (how long they have been employed, their work frequency, leaves, etc.), which can be mapped with the order and timing of their activities to understand user profiles. If the purpose of using my work is improving training sessions, the organisation can also provide information about the nature and content of the current training sessions. I built up a very good collaboration with the organisations which I visited during my study, due to the help of my study supervisor. The call centre

provided me with the statistics, information on user profile and training material.

10.2.2 Issues Related to Building the Models

One of the challenges of cognitive architectures is their transfer from research projects to commercial use, due to the difficulty of building the models [128]. This difficulty is also present for the application implementing my modelling approach. In particular, using it to build the inputs for the models requires knowledge of Java, XML and another programming language for the input generating script. It can be difficult for someone who is new to my application to build the inputs for LTM, goals and device notions due to the rules governing what can go into each component, the need for validity in the relationship between components (e.g. that a concept's 'relations' component refers to other existing concepts), and the length of the inputs for complex modelled work. To make writing inputs for the models easier and less time consuming for modellers who are not familiar with programming languages, or with the requirements for input formats, **I propose the development of a modelling language for building *input models* graphically within a graphical user interface (GUI)**¹. The modeller could just fill in some fields with information needed for the input models, and the underlying modelling language could build the input models. I also propose the development of a **model transformation to generate inputs for the user and device models in the required programming languages, out of the thus obtained input models**. The GUI could also offer the option to query input models, for being able to reuse inputs in new models, instead of writing them as new inputs for the models from scratch. My view is that, this way, parts of former models could be reused, which would speed up model development.

10.2.3 Issues Related to Generating the Results

Please be reminded that the outputs of repeated simulation runs of the models created by using my modelling approach are time logs, a log of the summary of steps performed in achieving intentions and goals by the simulated user, the relative frequency distribution of the total times after each run and, if the modeller opted for it, a log of all of the steps performed by the user and device. Having this in mind, the following are possible improvements for making the most of these results:

¹Please note the distinction between models for the user and device, and the proposed graphical representation of models of their inputs in this modelling language

- **Displaying the results centrally, within the GUI (described in the previous section),** to facilitate their retrieval
- **Adding to the GUI the option to search through data logs:** For the time being, the modeller needs to manually search through large logs containing the summary or detailed steps, to understand causes for user behaviour, or reasons for warning or error messages. To make this easier, I propose a solution such as a search facility, with different search criteria, for querying the logs. For example, entering the name of a goal, skill or intention could return the section from the log where it was addressed. Also, entering the name of the warning or error message could return the section of the log having led to it. Such a facility could be especially useful for software developers using my approach, who could search through the logs to better explore user mental models.
- **Adding to the GUI the option to search through time logs:** Similarly to above, understanding individual goal or intention durations requires a manual search through the time logs, and this could be made easier by a search facility.
- **Also generating other types of statistics, by using the times for goals and intentions**

10.2.4 Potential Extensions to the Modelling Approach

As also highlighted in Chapter 6 and Appendix E, there are some ways in which the modelling approach could be extended for use in different scenarios:

- **Investigating other factors which could potentially affect efficiency, and extending the modelling approach to also consider those which do:** As explained in Chapter 6, I have considered as inputs for the modelling approach either inputs which were also used in ICARUS (for the user model [139, 67, 137, 140, 142]) and PUM (for the device model [54, 46, 47]), or other inputs which I have added for allowing the simulated user to remember or forget information, and for automatically generating the information needed by the models during the simulation run. However, other factors, for example having to do with fatigue, attention, personality traits, or the work environment, could also affect a user in her work. I propose as future work, first of all, the collaboration with a psychologist, and the conduct of experiments to decide on a more complete list of factors which

would affect the efficiency of the system and work process. Secondly, I propose the extension of the modelling approach to consider more of the identified factors as inputs. In deciding just how many to consider, a compromise would need to be found between keeping the approach as economical and easy to use, on one hand, and obtaining reliable enough efficiency predictions, on the other.

- **Adding an improved perception mechanism, which can cater for issues such as eye movement, focus, attention, information scent, etc.:** As mentioned in Chapter 6 and Appendix E, the current perception mechanism is very simplistic, allowing the simulated user to see from the device everything that was rendered as on screen and visible. I have kept it as separate from the main user behaviour though, and it can easily be replaced by a different perceptual module. I propose as future work the collaboration with a psychologist for this purpose.
- **Evaluating and improving the mechanisms for remembrance/forgetfulness:** I have developed the simulated user's mechanisms for remembrance/forgetfulness as departing from a series of assumptions or conventions. I propose as future work their validation and evaluation against real human behaviour, as well as their further development, with the help of a psychologist. The investigation of remembrance/forgetfulness mechanisms like the ones from ACT-R ([34, 35]) would also be beneficial.
- **Considering user movement times as separate to system response times:** For my examples, I considered user movement times within the times of device actions, because it was difficult to observe them from videos of the computer screen, and due to NHS restrictions I could not also film the users. For studies where observing users is permitted, I propose a separate consideration of movement times, by implementing one of two solutions. The first solution would be to add them as a separate component to the user's device skills, the value of which would need to be provided as an input by the modeller. In this case, their time would be added to the simulated user's time before she sends the device skill's action to the device. The second solution would be to compute movement times, depending on the type of the user action, on the fly before adding them to the simulated user's time, by using the times from KLM ([61]).
- **Adding the option for the modeller to input static beliefs:** In my modelling approach, beliefs are inferred anew within each cognitive cycle. To represent

previous beliefs, which the simulated user should not forget, such as her knowledge about the patient profile or preferences, I propose the use of static beliefs as in ICARUS. As for LTM constructs and goals, they could be input for the user model by the modeller.

- **If novice users with little experience should be modelled, considering problem solving as in ICARUS:** The users whom I have modelled had enough experience with their routine work on the system, and therefore implementing problem solving was not necessary. If users who may have difficulties with the system are to be modelled (i.e. uncertainty is an issue), the modelling approach should be extended with problem solving behaviour.
- **Considering that goals can be interrupted and resumed:** A goal not being accomplished once all of the intentions have been addressed returns an error message to the modeller in my approach. This does not cater for cases where a real user might temporarily tackle another goal, before returning to the current one (e.g. monitoring another patient if the current patient has not picked up the phone). I therefore propose adding the possibility of temporarily interrupting a goal, and later resuming it.
- **Extending the modelling approach for predicting user errors in using a telehealthcare system:** Errors are an important concern in medical informatics, as they can endanger patients' lives. My modelling approach could be extended to compare the outputs of user work with expected outputs, to identify user errors. This could be done, for example, by using the list of configurations for user workload to automatically generate, apart from inputs for the model runs, also expected outputs to user work. Then, as the simulated user reaches each output, it could be compared with the expected outputs to detect a potential error.

10.3 Conclusions, Research Contributions

Large-scale telehealthcare pilots have the potential to prove the benefits of telehealthcare for improving patient care and costs, but are confronted with high risks due to deployment heterogeneity. Being able to predict their success, of which usability and, in particular, efficiency, should be important measures, is desirable yet difficult and costly. This thesis proposes the automation of efficiency evaluation, through the use of

modelling and simulation techniques, for making it possible to obtain system and work process efficiency predictions at lower costs.

The thesis has made the following important contributions:

- A generic methodology for reusing models which have been found to accurately predict certain characteristics of usability of a telehealthcare system and work process within a reference deployment, for making similar predictions in potential, real or hypothetical, deployments.
- A modelling approach which can be used alongside the methodology for obtaining efficiency predictions in each deployment. The approach builds user and system models and runs them in parallel to predict the relative frequency distribution of the total times spent by the user in her work, and traces of the performed steps. User models are inspired by, and extending, the ICARUS cognitive architecture. System models are represented as basic LTSs.
- An understanding of how data, and qualitative observations, could be collected from a real deployment context for instantiating models.
- An understanding of how the methodology and modelling approach could be used to explore the effects on system and work process efficiency of different user experience, ways of working, layouts of the work environment, system design, and foreseeable workload changes. Such investigations could be a valuable tool towards reducing costs in time and resource allocation, motivating investment in deployments or changes to a current deployment, and mitigating risks. I have showcased their use for a managerial scenario, but they could also represent an asset for investors or telehealthcare software developers.

To my knowledge, techniques like the ones proposed by this thesis have never before been used within a telehealthcare context. My work highlights the great potential of modelling and simulation for aiding in the scaling up of telehealthcare. At the same time though, the work is generic enough to also be used within other contexts.

10.4 Recommendations for Policy Makers

I have thoroughly enjoyed the time that I have spent in the company of COs, clinicians and their managers while conducting the usability study which has helped me collect

data for showcasing my approach in this thesis. The teams whom I have visited for the study have all been very accommodating and helpful, and they have all agreed to contribute to the study and to share with me experience and documentation. The one important limitation to my work was given by the type of permission that I could obtain from the NHS for conducting the study, and the time and effort in obtaining it. Even with the help of my PhD co-supervisor and my usability study supervisor, it took me 2 months to obtain an observer contract. The problem was deciding on the type of research that I was conducting and whether it required ethical approval, on one hand, and getting the right permission for my status as a PhD student in Informatics, on the other. The fact that I was not a medical practitioner or healthcare researcher made things worse, and we were referred from one NHS department to another. In the end, the restrictions imposed by my observer contract meant that I could not video record NHS staff (but just their computer screen) while working, which limited the amount and accuracy of data that I could collect, and thus the reliability of my results. Moreover, due to patient confidentiality reasons, I needed to analyse all of the data on NHS premises.

I think that bureaucracy and the risk averse culture of the NHS are strong barriers to health IT research and the diffusion of innovation. A greater openness to collaboration with Informatics researchers and engineers is required for addressing the critical threat posed by demographic changes. In particular, I strongly believe that researchers should have more access to clinicians, the systems and external tools that they use. One good solution for this would be, from my point of view, having a separate ethical research committee for health IT research. At the time being, although this type of research is concerned with service delivery, and thus information on patient condition and their treatment is not its focus, it undergoes the same level of ethical scrutiny as medical research.

Appendices

Appendix A

Information Sheets

A.1 Provided to NHS 24 COs

Background

My name is Cristina-Adriana Alexandru and I am a PhD student in the Laboratory for Foundations of Computer Science in the University of Edinburgh School of Informatics. As part of my PhD, I am interested in facilitating the work of healthcare practitioners with telemedicine systems by making these systems more easy to use, user friendly and conformant to user requirements (improve their usability). To this end, I am building techniques and tools which can predict the types and effects of the usability problems which healthcare practitioners encounter in telemedicine systems such that these problems could be more timely addressed. I am currently exploring the possible usability problems of the [NAMED] system, which you are using for remote patient monitoring.

What I would like from you

I would like to observe you while performing your routine patient monitoring work using the [NAMED] system. This would help me understand how you would like the system to work better, and how the carious tasks you perform might be made easier. The observation would also help me understand what other tasks you need to perform in relation to monitoring (e.g. contacting other people for advice, phoning patients) as well as other programs you have to use or written recordings you may need to make and how much time this all takes. The observation will not exceed an hour. With your permission, I would like to video record the computer screen during the observation (you will not appear on the video), but if you do not agree I would just take notes

instead.

After the observation, I would like to interview you briefly in order to understand your general opinion about the system and its different functionalities, its pros and cons, any problems you have encountered while using it and any suggestions you may have for improving it. The interview will only focus around those functions of the system that I have observed you using, and of which you may have opinions. I realise how busy you are and so the interview will take at most another half an hour. With your permission, I would like to audio record the interview. However, if you are not happy with this, I would only take notes.

Please rest assured that all your responses will be kept confidential. Your name or clues leading to your identity or personal information will not be given at any stage. Moreover, you are free to end the observation or interview sessions at any time. The final report, containing the results of the observation and interview analysis and drawn conclusions will be made available to you on request.

I would be happy to answer any questions regarding this information sheet. Thank you for the time taken in reading it.

A.2 Provided to Reparticipating NHS 24 COs

Thank you for kindly accepting to contribute to the first phase my study by being observed and interviewed regarding the usability of the previous version of the [NAMED] telemonitoring system. I am now conducting a second part to this study, in which I would like to understand what the current version has changed in terms of its ease of use, user friendliness and conformance to your requirements.

What I would like from you

With your permission, I would like to observe you while performing your routine patient monitoring work using the new version of the [NAMED] system. I would be interested to see how elements such as new website functionality that you may use, the fact that you now use Google Chrome as your browser, or any changes to the interface and to the underlying protocol and process, affect the time you spend doing your monitoring work and the process and tools you use. The observation will not exceed one hour. With your permission, I would like to video record the computer screen during the observation (you will not appear on the video), but if you do not agree I would just take notes instead.

After the observation, I would like to interview you very briefly (for no more than 20 minutes) in order to understand your opinion about the changes made to the system and the underlying protocol and process, their pros and cons, any new problems you have encountered while using them and any suggestions you may have for improving them.

With your permission, I would like to audio record the interview. However, if you are not happy with this, I would only take notes.

Please rest assured that all your responses will be kept confidential. Your name or clues leading to your identity or personal information will not be given at any stage. Moreover, you are free to end the observation or interview sessions at any time. The final report, containing the results of the observation and interview analysis and drawn conclusions will be made available to you on request.

I would be happy to answer any questions regarding this information sheet. Thank you for the time taken in reading it.

A.3 Provided to Care Providers (System Version 1.2)

Background

My name is Cristina-Adriana Alexandru and I am a PhD student in the Laboratory for Foundations of Computer Science in the University of Edinburgh School of Informatics. As part of my PhD, I am interested in facilitating the work of healthcare practitioners with telemedicine systems by making these systems more easy to use, user friendly and conformant to user requirements (improve their usability). To this end, I am building techniques and tools which can predict the types and effects of the usability problems which healthcare practitioners encounter in telemedicine systems such that these problems could be more timely addressed. I am currently exploring the possible usability problems of the [NAMED] system, which you are using for remote patient monitoring.

What I would like from you

I would like to observe you while performing your routine patient monitoring work using the [NAMED] system. This would help me understand how you would like the system to work better, and how the various tasks you perform might be made easier. The observation would also help me understand what other tasks you need to perform in relation to monitoring (e.g. contacting other people for advice, phoning patients) as well as other programs you have to use or written recordings you may need to make and how much time this all takes. The observation will not exceed an hour. With your permission, I would like to video record the computer screen during the observation (you will not appear on the video), but if you do not agree I would just take notes instead.

After the observation, I would like to interview you briefly in order to understand your general opinion about the system and its different functionalities, its pros and cons, any problems you have encountered while using it and any suggestions you may have for improving it. The interview will only focus around those functions of the system that I have observed you using, and of which you may have opinions. I realise how busy you are and so the interview will take at most another half an hour. With your permission, I would like to audio record the interview. However, if you are not happy with this, I would only take notes.

Please rest assured that all your responses will be kept confidential. Your name or clues leading to your identity or personal information will not be given at any stage.

Moreover, you are free to end the observation or interview sessions at any time. The final report, containing the results of the observation and interview analysis and drawn conclusions will be made available to you on request.

I would be happy to answer any questions regarding this information sheet. Thank you for the time taken in reading it.

Appendix B

Consent Form

Participant Identification Number for this study:

CONSENT FORM

Title of Project: On the Design and Validation of ICT for Medical Domains; Evaluation of the [NAMED] Telehealth System Version 1.2

Name of Researcher: Cristina-Adriana Alexandru

1. I confirm that I have read and understand the information sheet dated ☐ for the above study and have had the opportunity to ask questions
2. I understand that my participation is voluntary and that I am free to withdraw at any time, without giving any reason and without my legal rights being affected. ☐
3. I agree to take part in the study. ☐
4. I agree for the observation to be video recorded ☐
5. I agree for the interview to be audio recorded ☐

Name of Participant

Date

Signature

Researcher

Date

Signature

Cristina-Adriana Alexandru

Appendix C

Interview Schedules

C.1 Questions for NHS 24 COs

The website

I would first like to ask you some general questions regarding your experience with the [NAMED] telehealth website:

1. How long have you been using the [NAMED] website?
2. How often do you use the website? How many hours do usually spend using it in a day?
3. What sort of training have you received before starting to interact with the [NAMED] website?
4. What was your initial reaction to the website regarding the level of difficulty in using it?
5. Was the website's functionality easy to learn (after having received training, if any)? How long did it take to learn? How does it compare to any other telehealth systems you have used?

Next, I would like to ask you some general questions regarding the website's usability:

1. You mainly use the Monitor, the Alerts and the Patient Monitoring pages (or other). Do you ever use any of the other screens (e.g. Patient details from homepage, Pathways from Alerts?) If so, in what circumstances and how?

2. I noticed that you use the spreadsheet and call back sheet often. Would you like them to be included in the website? How would you like this to work?
3. Do you ever write anything on paper? Would you like this to also be included in the website? How would you like this to work?
4. Is there any additional functionality which, if added, could make your work easier?
5. (if answer to 1 is yes') Could you provide an example of a command you accomplish rarely and tend to forget how to accomplish? In case you forget, how easy is it to get back on track? Is there any available documentation to help you?
6. How happy are you with the website's presentation of information? Are there any parts which come to mind as having irritated you? What do you like best/worst about the website?

In the next part of the interview I would like us to discuss about the different pages that you use in the website:

1. Specific questions about the observed process- why do you do ...? What do you look at ...?
2. In the homepage, is the information provided on each patient useful? Is there any other information you feel would be appropriate on the patients' list?
3. Are the buttons for each patient clear and useful? Would you have it any other way?
4. Do you ever use the Patient details' button in the homepage? Would this ever be useful?
5. What do you look at in the Monitor page? Do you have any comments on how the information is represented here? Would you have it any other way?
6. In the Alerts page, is the way each alert is represented useful? Suggestions, other info?
7. Do you have any comments on the functionality of closing down alerts?
8. In the Patient Monitoring page, what do you look at and in what order? Would you prefer it to be some other way?

9. Do you ever need to look back into an already checked patient? If so, how is it that you search for the patient?
10. Generally, do you consider that your routine work on the website takes the right amount of time and effort (in terms of number of clicks)? Please comment and provide suggestions.

As related to results from diary method and any other technical problems:

1. Did you ever encounter system delays (e.g. system getting stuck/taking long time to load)? If so, how frequent and how long are they? On what would you blame them?
2. Did you ever come across impossible readings/incorrect alerts? If so, how frequent were they? What was the criticality of their outcomes on your monitoring activity?
3. Did you ever encounter other technical issues?
4. In case of technical issues on the system, what kind of support do you have available?

The Process

In the last part of the interview, I would like to ask for your opinion about the processes behind your monitoring activities: the protocol and call back criteria.

1. Is the protocol easy to follow and easy to understand? Does the home monitoring recording tool (the spreadsheet) help you with using it (any suggestions if not enough)?
2. From your observations of the patients' number of alerts, do you feel there are any areas where the protocol could be improved? Do you have any specific suggestions?
3. From your experience in talking with patients, what is your opinion about the effectiveness of the questions asked? Do you have any suggestions for improving on this?
4. Does the spreadsheet help you enough for deciding on the order of questions while you are on the phone with the patient? If not, please provide suggestions on how it could be improved.

5. Have you adapted the order in which you ask questions or the questions themselves in any way to better fit patient needs?

Are there any more observations or comments you would like to add?

Thank you for your time! Please feel free to request the final report from me at the end of August, when the study will have been completed.

C.2 New Questions for Version 1.2 for NHS 24 COs

1. Thinking only about the website, and not about the protocol that you use, have you observed any important changes in your work as related to the new version of the website? Please explain which and why.
2. In particular, what is your opinion about the new way of displaying measurements in the patient monitoring page (no decimals, table first)?
3. In the Alerts' page, do you ever need to search for alerts which are older than today? If so, how do you do this (pagination/search)? Have you used the alerts search functionality? If so, what is your opinion about it?
4. Do you ever need to change a patient's data in programme details? If so, have you noticed how this process has changed? What is your opinion about it?
5. Have you encountered any new technical issues?
6. What is your opinion about the new protocol? How is it different from the previous one?
7. Is the new protocol easy to follow and easy to understand? Does the spreadsheet help you enough with using it (any suggestions if not enough).
8. From your observations about alerts and monitoring outcomes, do you feel there are any areas where the protocol could be further improved? Do you have any specific suggestions?
9. What is your opinion about the new call back criteria? How is it different from the previous one?
10. From your experience in talking with patients, what is your opinion about the effectiveness of the new questions asked? Do you have any suggestions for improving on this?
11. Does the spreadsheet help you enough for deciding on the order of questions while you are on the phone with the patient? If not, please provide suggestions on how it could be improved.
12. Have you ever needed to adapt the order in which you ask questions or the questions themselves in any way to better fit patient needs?

Are there any more observations or comments you would like to add?

Thank you for your time! Please feel free to request the final report from me at the end of August, when the study will have been completed.

C.3 Questions for Care Providers

The Website

I would first like to ask you some general questions regarding your experience with the [NAMED] telehealth website:

1. How long have you been using the [NAMED] website?
2. How often do you use the website, in general, and after the introduction of the new version? How many hours do usually spend using it in a day?
3. What sort of training have you received before starting to interact with the [NAMED] website?
4. What was your initial reaction to the website regarding the level of difficulty in using it?
5. Was the website's functionality easy to learn (after having received training, if any)? How long did it take to learn? How does it compare to any other telehealth systems you have used?

Next, I would like to ask you some general questions regarding the website's usability:

1. Please briefly take me through the way that you make a decision by using the [NAMED] system. Do you consider both emails that are for information and those that are for action? What kind of measurements/surveys/patient reactions on the phone alert you?
2. You mainly use the Alerts and the Patient Monitoring pages (or other). Do you ever use any of the other screens (e.g. Monitor, Patient details from homepage, Pathways from Alerts?) If so, in what circumstances and how?
3. I noticed that you use the spreadsheet and email quite often. Would you like them to be included in the website? How would you like this to work?
4. Do you ever write anything on paper? Would you like this to also be included in the website? How would you like this to work?
5. Is there any additional functionality which, if added, could make your work easier?

6. (if answer to 1 is yes') Could you provide an example of a command you accomplish rarely and tend to forget how to accomplish? In case you forget, how easy is it to get back on track? Is there any available documentation to help you?
7. How happy are you with the website's presentation of information? Are there any parts which come to mind as having irritated you? What do you like best/worst about the website?

In the next part of the interview I would like us to discuss about the different pages that you use in the website:

1. Specific questions about the observed process- why do you do ...? What do you look at ...?
2. Thinking only about the website, and not about the protocol that you use, have you observed any important changes in your work as related to the new version of the website? Please explain which and why.
3. In the homepage, is the information provided on each patient useful? Is there any other information you feel would be appropriate on the patients' list?
4. Are the buttons for each patient clear and useful? Would you have it any other way?
5. Do you ever use the Patient details' button in the homepage? Would this ever be useful? (if used 6. What do you look at in the Monitor page? Do you have any comments on how the information is represented here? Would you have it any other way?)
6. In the Alerts page, is the way each alert is represented useful? Suggestions, other info?
7. In the Alerts' page, do you ever need to search for alerts which are older than today? If so, how do you do this (pagination/search)? Have you used the alerts search functionality? If so, what is your opinion about it?
8. Do you have any comments on the functionality of closing down alerts?
9. In the Patient Monitoring page, what do you look at and in what order? Would you prefer it to be some other way?

10. What is your opinion about the new way of displaying measurements in the Patient Monitoring page (no decimals, table first)
11. Do you ever need to change a patient's data in Programme Details? If so, have you noticed how this process has changed? What is your opinion about it?
12. Do you ever need to look back into an already checked patient? If so, how is it that you search for the patient?
13. Generally, do you consider that your routine work on the website takes the right amount of time and effort (in terms of number of clicks)? Please comment and provide suggestions.

As related to technical problems

1. Did you ever encounter system delays (e.g. system getting stuck/taking long time to load)? If so, how frequent and how long are they? On what would you blame them?
2. Did you ever come across impossible readings/incorrect alerts? If so, how frequent were they? What was the criticality of their outcomes on your monitoring activity?
3. Did you ever encounter other technical issues?
4. In case of technical issues on the system, what kind of support do you have available?

The Process

In the last part of the interview, I would like to ask for your opinion about the processes behind your monitoring activities: the protocol and call back criteria.

1. Are you aware of the call back criteria and protocol?

If so:

1. What is your opinion about the new call back criteria? How is it different from the previous one?
2. From your observations of the patients' alerts and state, do you feel there are any areas where the call back criteria could be further improved? Do you have any specific suggestions?

3. What is your opinion about the new protocol? How is it different from the previous one?
4. What is your opinion about the effectiveness of the questions asked for helping you decide on the patient's state? Do you have any suggestions for improving on this?

If not, for assessment questions and questions to the patient in turn:

1. Have you noticed any changes to the questions in the spreadsheet since the new version of the system? How effective are they in general and as compared to the system's previous version for helping you understand the patient's state?
2. Are there any questions which, if added, could help you more on deciding on the patient's state?
3. Are there any questions which you believe should be reformulated?

Are there any more observations or comments you would like to add?

Thank you for your time! Please feel free to request the final report from me at the end of August, when the study will have been completed.

Appendix D

List of Themes and Subthemes

D.1 For the First Part of the Study

1. Demographics

- Participant experience with the [NAMED] system
- Duration of use in a day

2. Support

- Documentation
- Help at hand
- Training

3. Routine HM work

- Use of the system
- Use of external tools

4. Design

- **Navigation**
 - General opinions
 - Navigation in between applications
 - * General opinions
 - * System- email
 - * System- statistics table
 - Information retrieval

- * Retrieving repeated readings
- * Retrieving thresholds
- * Retrieving reading trends
- * Retrieving patients to monitor
- * Retrieving differences in survey score
- * Retrieving date and time of last readings
- * Retrieving patient demographics
- * Retrieving all reading types

- **Presentation**

- General opinions
- Information on patients
 - * Readings in Monitor page
 - * Readings in Patient Monitoring
 - * Computed alerts
- Buttons
 - * Placement of buttons
 - * Button names on homepage
- Key information
- Demographics form on Programme Details

- **Interaction**

- Ease of use
- Learning

5. Functionality

- Technical issues
- Comments on particular functionality
 - Closing down alerts
 - Filling in key information
 - Changing patient/programme details
- Suggestions for additional functionality
 - Integration between the website and the Excel spreadsheet
 - Integration between the website and the email facility
 - Integration between the website and the stats table

6. Protocol and call back criteria

- Ease of use
- Logic
- Order of questions
- Patient question effectiveness
- Effectiveness of calling patient
- Effectiveness of soliciting clinician
- Effectiveness of emailing care providers

D.2 For the Second Part of the Study

1. Demographics

- Participant experience with the [NAMED] system
- Use of other computer systems and approaches for telehealth monitoring

2. Communication within the project

- Support
 - Training
 - Documentation
 - Contacts for technical or process issues
- Communication between weekday and weekend teams

3. Design

- **Navigation**
 - Navigation within the website:
 - * General opinions
 - * Searching for a patient on the homepage
 - * Moving in between patient pages
 - * Saving changes in Programme Details
 - * Retrieving patient thresholds
 - * Retrieving history information on alerts
 - * Retrieving the full details of patient surveys
 - * Making changes to key information
 - * Other comments on navigation
 - Navigation in between applications
 - Suggestions for improving navigation:
 - * Links between the applications constituting the system
 - * Links between the patient pages on the [NAMED] website
 - * ‘Thresholds’ button on the homepage
- **Content**
 - Suggestions for additional content
 - * Contact details for the patient’s GP on the homepage or Programme Details page

- * Medical information for the patient on the Programme Details page
- * More accessible patient thresholds
- * Other suggestions of content

- **Presentation**

- Data representation
 - * Patient list on the homepage
 - * Readings in the table representation from the Patient Monitoring page
 - * Visualisation preference in the Patient Monitoring page
 - * Readings in the chart representation from the Patient Monitoring page
 - * Surveys in the Patient Monitoring page
 - * Alerts in the alerts page
- Page layout
 - * The layout of the homepage
 - * The layout of the Patient Monitoring- Readings page
 - * The layout of the Patient Monitoring- Surveys page
 - * The layout of the Alerts page
- Buttons
 - * Button names on homepage
 - * Name of the View history button from the Alerts page
 - * Placement of the Back button from the website
 - * Placement of the View surveys button on the Patient Monitoring page

- **Interaction**

- Learning
- General opinions

4. **Functionality**

- **Access to functionality**
 - Editing key information for a suspended/reinstated patient
 - Changing patient thresholds
- **Comments on particular functionality**
 - Changing time intervals for readings in Patient Monitoring
 - Making a decision on alerts
 - The View history functionality for alerts

- The functionality behind forms for patient information in Programme Details
 - Filling in key information on the Programme Details page
 - Saving changes in Programme Details
 - The functionality of the browsers Back button
 - Sorting patients on the homepage
- Suggestions for additional functionality
 - Integration between applications
 - Team specific website interfaces
 - Making a decision on all alerts automatically when making a decision on one in the Alerts page
 - Notes facility
 - Automatic statistical reports
 - Other suggestions for additional functionality
- Technical issues

5. Process

- Quality and usefulness of the information from NHS 24 and the logic behind it
 - The alert emails received from NHS 24
 - The notes made by the NHS 24 call operators on the spreadsheet
 - The answers to questions from the rest of the spreadsheet
- Effectiveness of patient questions
- Meeting purposes
 - Promoting self-management
 - Reducing workload for care providers
 - More timely care for patients

Appendix E

Detailed Description of the Improved Modelling Approach

E.0.1 Notions for the User Model

E.0.1.1 The Perceptual Buffer

The perceptual buffer is an STM construct consisting of a set of **percepts**, which represent the elements which the simulated user sees (perceives) from the device on the current cognitive cycle (similar to ICARUS [139, 67, 137, 140, 142]), or, in contrast to ICARUS, which she remembers from previous cycles. The main components of the percepts are presented in Fig. E.1, and I will explain each of them in detail.

The **type** component is used, as in ICARUS, to distinguish between different types of information that a real user could perceive from a system's screen. For example, a user of a telemonitoring website could perceive percepts of type "page", "table", or "button", "patient", "reading". Please note that conceptual distinction between these types, which I will explain below.

I have extended the percept notion from ICARUS by adding a **location** component, which is used to distinguish between the percepts which are seen on the different applications which may be used as part of the telehealthcare 'system'. For example, we considered that the telehealthcare system described in Chapter 3 consists of all of the applications which are used as part of the service: the telemonitoring website, two Excel spreadsheets and an email client, to which we could also add the users' use of the operating system (OS). If we considered the modelling of the use of the whole 'system', we could thus use the following locations for the percepts: "website", "spreadsheet1", "spreadsheet2", "email_client", "OS".

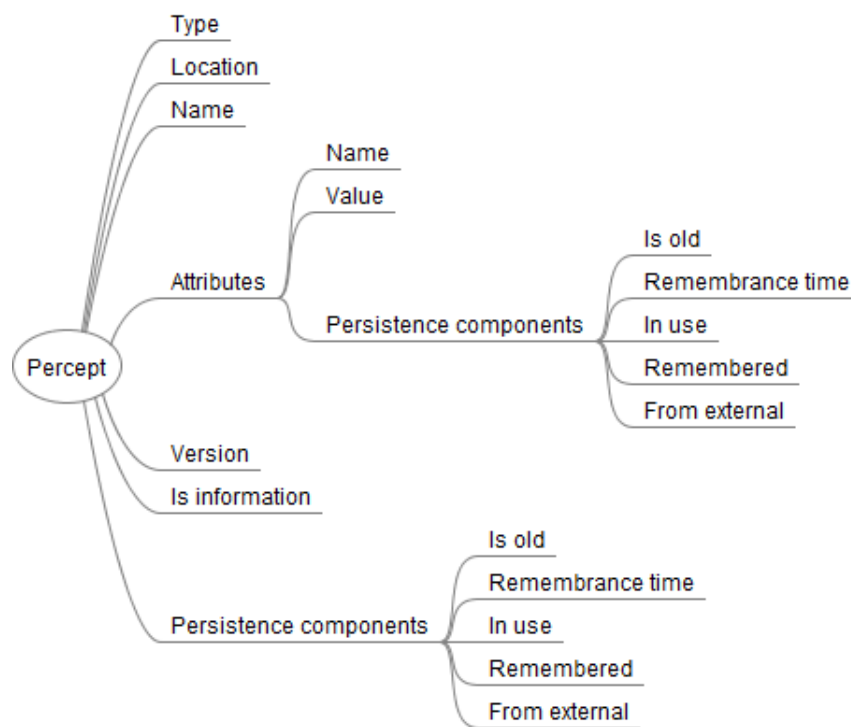


Figure E.1: Percept components

The **name** component identifies the percept. As in ICARUS, names for all of the percepts which are seen by the simulated user from the device within each cognitive cycle must be unique. For this purpose, I used names in a similar way to how ids are used to identify each element of a table in a database. They consist of the first letter of each percept's type, or two letters if there are several types starting with the same letter, and a unique number. Consecutive numbers are used if the simulated user is to perceive more percepts of the same type. For example, if the simulated user sees 10 percepts of type "percept" at one time on the device, their names would be "p1", "p2", ..., "p10". At the same time though, in contrast to ICARUS, I decided to use the same name for percepts placed on different system states, which would be perceived by the simulated user within different cognitive cycles, if they refer to the same conceptual notion. For example, the simulated user would be aware, when navigating between the different pages of a patient on a telemonitoring website, that information about the same patient is provided, and therefore she would perceive percepts with the same name, e.g. "p1"¹. Additionally, remembered percepts with the same name could coexist with currently perceived ones within the simulated user's perceptual buffer.

¹Please note though that this does not make them the same percepts, as another of their components, their version, which will be described below, might differ

Percepts optionally have a set of **attributes**. As in ICARUS, attributes have names and values. An attribute's **name** corresponds to a type of information that is provided about the percept on a certain cognitive cycle, on a system state. For example, while on the homepage a simulated user could see for the percept of type "patient" information about the patient's condition and whether the patient is highlighted as having triggered alerts (represented as two attributes with names "condition" and "highlighted"), on a Details page she could see for a percept with the same type information about the patient's full demographics (e.g. attributes with name "first name", "surname", "date of birth", "address", etc.). The **value** of the attribute with that name can be numeric or literal, as in ICARUS, and must be atomic (i.e. cannot have subvalues). For example, for an attribute with name "age" the value could be "50", and for an attribute with name "first name" the value could be "Helen". In contrast to ICARUS, values in my approach cannot reference the name of subpercepts. This would be useful for representing the relationship between percepts, e.g. to show that a percept of type "reading" belongs to a percept of type "patient" named "p1", by having an attribute named "belongs_to" for the percept of type "reading" and the name of the percept of type "patient", "p1" as the attribute's value. I am currently representing such relationships only through the use of concepts, as will be described below. I believe that referencing the name of subpercepts in values would nevertheless be a useful extension to my modelling approach, and therefore propose it as future work.

I have extended the notion of attributes from ICARUS by adding to attributes in my approach components which help in the remembrance/forgetfulness processes for the user, which I name '**persistence components**' and which will be further described in the 'Behaviour' section:

- **Is old:** a truth value (TRUE/FALSE) indicating whether the attribute is one which was seen by the simulated user earlier (TRUE), or has just been seen (FALSE)
- **Remembrance time:** a numeric value indicating for how long the attribute has been remembered by the simulated user
- **In use:** a truth value (TRUE/FALSE) indicating whether the value of the attribute is currently used by the simulated user within a skill, or required by her to be used within a future skill
- **Remembered:** a truth value indicating whether the value of the attribute was required by the user to be used within a future skill

- **From external:** a truth value indicating whether the value of the attribute is to never be forgotten, as having been set up by a simulated user's action outside of the system (e.g. the simulated user writing the attribute on paper)

A percept's **version** is an extension to the definition of percepts from ICARUS. It is a number used to uniquely identify all of the different percepts with the same name from the system. While names help in the modelling of percepts which are conceptually the same notion, versions help in identifying exactly the percepts with a certain name which were used by the simulated user in her skills, and which might have incurred slight changes as an effect to the skill, as will be explained in the next section. The name-version pair therefore uniquely identifies percepts.

A percept's **'is information'** component is also an extension to the definition of percepts from ICARUS. It is a truth value indicating whether the percept is referring to structural, or to information-type elements on the system. The difference between percepts referring to structure and percepts referring to information is the conceptual distinction that I was referring to when describing the 'type' component of percepts. From the examples given there, "page", "table" and "button"- type percepts are structural-type percepts, as they refer to the structural components of the system, and their 'is information' component would be set to FALSE. We could see structural-type percepts as making up the layout of the system, in which information would be fit in. The other percepts, with type "percept" and "reading" are information-type percepts, as they represent the information that would populate the system's structure, and which would need to be retrieved by the system from its databases. For such percepts, the 'is information' component would be set to TRUE.

As their attributes, percepts can also have **persistence components**, as an extension to percepts from ICARUS. Their meaning is similar to that of attributes, the distinction being that they are only set up if the percept is to be remembered or used by itself, with no attributes. Apart from the 'is old' and 'remembrance time' components, they do not have any relationship with the components of the same type from attributes. The value of the **'remembrance time'** component cannot be greater than the minimum value of the components of the same type from attributes, but it can be smaller. The value of a percept's 'is old' component will always correspond with that of its attributes' component with the same name.

Example: A possible percept that a user could observe from a telemonitoring website is a patient with her characteristics. I am neglecting in this example the persistence components, the setting up and use of which I am going to explain in the

section about the remembrance/forgetfulness behaviour:

(patient website p1 ((name Helen) (condition COPD) (highlighted red)) 1 TRUE)

This percept therefore has the type “patient”, the location of the patient is on the website, its version is 1 (i.e. it is the first, or only, percept with the name p1), and her attributes have the names “name”, “condition”, and “highlighted” (to indicate whether she is alerting) with the values “Helen”, “COPD” and “red”. Moreover, as the percept is of type “patient”, and therefore represents information that would be populated onto the telemonitoring website, its ‘is information’ component is set to TRUE. I will use notation similar to the one from above whenever describing percepts in future examples.

E.0.1.2 The Belief Memory

As in ICARUS, the belief memory in an STM construct which stores a set of **beliefs**, representing what the simulated user has deduced about the relationship between percepts, or the characteristics of a single percept. In my modelling approach, beliefs can be inferred by the simulated user regarding any of the percepts: those which were perceived by her in the current cognitive cycle (as in ICARUS), and those which were remembered by her from previous cycles (as different from it).

With respect to the structure of beliefs, I have kept the same components as were defined in ICARUS: a predicate which I called the belief’s **name**, and a list of **arguments** referencing percepts with the same name from the perceptual buffer.

Some beliefs from the belief memory are, as in ICARUS, inferred as instances of the more general *concepts* from conceptual memory, which will be described below. These beliefs should show relations between the different percepts seen by the simulated user from the telehealthcare system.

Examples: In a telemonitoring scenario, one could infer that a percept with type “patient” and name “p1” needs to be monitored, and represent this as a belief thus:
(needs_monitoring p1)

This belief therefore has the name “needs_monitoring” and the argument “p1”, where the argument matches with a percept with name “p1” from the perceptual buffer. This is an inferred property for the single percept p1.

Should the user be interested in finding all of the patients from the system who suffer from COPD, and have percepts for them uniquely named p1, p2, p3, etc., she

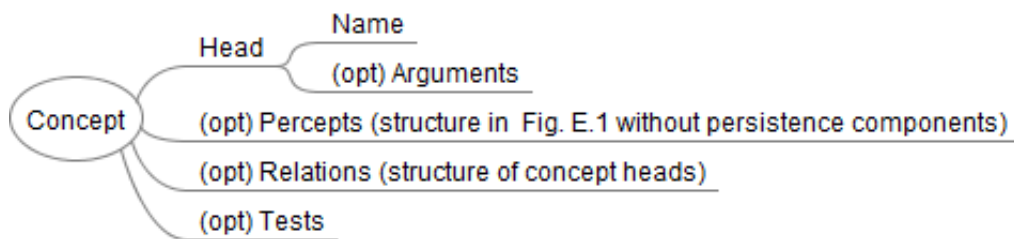


Figure E.2: Concept components

may infer that two patients suffer from the same condition and represent this as a belief:

(COPD_patients (p1 p2))

This is an example of a belief named “COPD_patients”, representing an inferred relationship between percepts, this time between the two percepts “p1” and “p2”.

In my implementation, other beliefs could also be added directly to the belief memory during skill execution, if they are the true effects of *external skills* (skills which are performed in the environment by the simulated user), which are assumed to always be achievable. Unlike in ICARUS, I have not considered the addition by the modeller of static beliefs, which do not need to be re-inferred. There are some cases where this could be useful, for example to describe the user’s prior knowledge about the patient’s profile, and therefore I am proposing adding this functionality as future work.

I will describe the notions of concepts, skills, and effects in the next subsections, and the functionality of inferring beliefs in different cases in the subsection on the behaviour of the modelling approach.

E.0.1.3 The Conceptual Memory

For clarity, I have decided to describe conceptual memory, an LTM construct, before the last type of STM construct, intention memory. Conceptual memory consists, like in ICARUS, of a set of **concepts**, which define the general inferences that a simulated user can make about the relationship between percepts or percept properties. I have kept the same components for concepts as were defined in ICARUS. They are represented in Fig. E.2, and I will describe each below.

Like in ICARUS, a concept’s **head** consists of a **name** which identifies the concept, and an optional list of **arguments** presented as variables, which represent the percept names on which the concept is applied. In the examples from below, for concepts and

skills, I am going to use ‘?’ at the beginning of variables to distinguish them from percept instances from the perceptual buffer.

A concept can have an optional set of **percepts** defining those percepts which must be present in the perceptual buffer such that the concept is applicable. In my approach these percepts and their attributes do not contain the persistence components. I also added here, as not specified by ICARUS, that **these percepts must always include a full specification of the head’s arguments (if any)**, and therefore this component must be defined if such arguments were provided. Only the attributes which are needed for the percepts in the concept (as described below) are provided, as unordered, for each percept. In contrast to ICARUS, the values of the attributes can only be variables themselves, to match against the values of the attributes of the percepts that match from the perceptual buffer, and they cannot also be literal or numerical values.

Also as optional, a concept can contain as a component a set of **relations** referencing the heads (name and arguments) of other concepts which must also hold/not hold if the name is negated (i.e. exist as instantiated beliefs in the belief memory or not) such that the current concept is applicable (as in ICARUS). They are defined as heads, consisting of a name and arguments which, in the case of my approach, must be variables which were defined in ‘percepts’. In ICARUS, they can also be new variables used with the meaning of ‘any’, but I considered that this was not necessary for my models.

Tests are another optional component of concepts. They represent logical or arithmetical expressions between variables, which must hold to make the concept applicable. In contrast to ICARUS, the variables need to have been defined as attributes of the percepts from the ‘percepts’ component, and could not also have been defined in ‘relations’. Also, my tests can include checks that an attribute takes a certain literal or numerical value, which in ICARUS was done by directly including the value of the attribute for the percept in the concept’s ‘percepts’ component.

I added here a convention as to what should go into the ‘percepts’ component of a concept: percepts are only needed in this component if they define the head’s arguments, are used as arguments themselves in ‘relations’, or their attributes are required in the concept’s ‘tests’ field. They are not needed if they are used in the definitions of the referred concepts from ‘relations’, as in this case they are checked by these concepts themselves.

As in ICARUS, concepts in my approach should include either a ‘relations’ component, a ‘tests component’, or both. Concepts which include a ‘relations’ component are non-primitive concepts, because they depend on other concepts (the ones defined in

‘relations’) for being applicable. They form a hierarchy of concepts and subconcepts. Concepts which only include a ‘tests’ component are primitive, and represent the leaves of this hierarchy.

Examples: The first belief about a patient needing monitoring which I provided as an example for the representation of beliefs can be an instance of the following concept:

Head: (needs_monitoring ?p)

Percepts: (patient_website ?p (highlighted ?h) 1 TRUE)

Tests: (== ?h red)

The concept is therefore named “needs_monitoring”, and it takes one argument referring to a percept of type “patient”, from the website, containing information, and having (at least) an attribute of type “highlighted”. The concept is applicable if the value of the percept’s “highlighted” attribute is “red”. Therefore, a patient is only considered as needing monitoring if she is highlighted on the website in red. The second belief about two patients both suffering from COPD can be an instance of the following concept:

Head: (COPD_patients (?p1 ?p2))

Percepts: ((patient_website ?p1 (condition ?c1) 1 TRUE) (patient_website ?p2 (condition ?c2) 1 TRUE))

Tests: ((== ?c1 ?c2) && (== ?c1 COPD))

This concept thus takes two arguments defined as percepts of type “patient”, and becomes applicable if both of the percepts have an attribute of type “condition”, the values of which are equal, and equal to “COPD”. Therefore, it is considered that two patients both suffer from COPD if their conditions are the same and equal to “COPD”. To also exemplify the use of relations for concepts, let’s say that the user is looking for those patients who suffer from COPD, but who also need monitoring. A useful concept would in this case add to the concept from above a check of whether the two patients also need monitoring. If the concept from the first example (needs_monitoring) is also present in the conceptual memory, the current concept could check that it holds for two patients by using it in its ‘relations’ component thus:

Head: (COPD_patients_needing_monitoring (?p1 ?p2))

Percepts: ((patient_website ?p1 (condition ?c1) 1 TRUE) (patient_website ?p2 (condition ?c2) 1 TRUE))

Relations: ((needs_monitoring ?p1) (needs_monitoring ?p2))

Tests: ((== ?c1 ?c2) && (== ?c1 COPD))

Please note that I have not added the attributes of type “highlighted” for the percepts of type “patient”, although they are required by the “needs_monitoring” concepts, as they would have already been checked by them. We could of course obtain the same belief as an instance of concept COPD_patients_needing_monitoring if all of the tests of this concept and those of the concepts from its relations field were all put together in this concept’s tests. The need for the separate “needs_monitoring” concept is motivated though by the assumption that beliefs about a patient needing monitoring would also be useful for the simulated user as derived independently.

E.0.1.4 The Skill Memory

As in ICARUS, the skill memory is an LTM construct which consists of a set of **skills**, representing the activities that the simulated user has available for pursuing her goals. As additional though, I have divided the general notion of skills that ICARUS has proposed into two categories for my purposes: the skills that the simulated user can perform on the device, which I call ‘device skills’, and the skills that she can perform in her static environment, separate from her work on the system, which I call ‘external skills’. As a convention, I consider that all of the activities taking place on a separate computer system, but in relation to the system that we are modelling, should be modelled as part of this system, while the environment (e.g. the simulated user’s office) should be seen as static. The conceptual difference between the notions of ‘device skills’ and ‘external skills’ is given by our interest in thoroughly modelling a user’s interaction with the system, and only keeping a simplified view of her interaction with her static environment. In particular, on one hand we are interested in the full requirements for a skill on the system (device skill), its resulting steps (subskills) or action, and effects, because the system might always change in an unexpected way for the user and we are interested in modelling that and the interaction. On the other hand, we can take a much more simplified view regarding external skills, and consider that they are always directly achievable and lead to the expected effects, and therefore even avoid the representation of the environment for the user. As the notions of device and external skills have conceptual differences, although I have implemented them together, I will

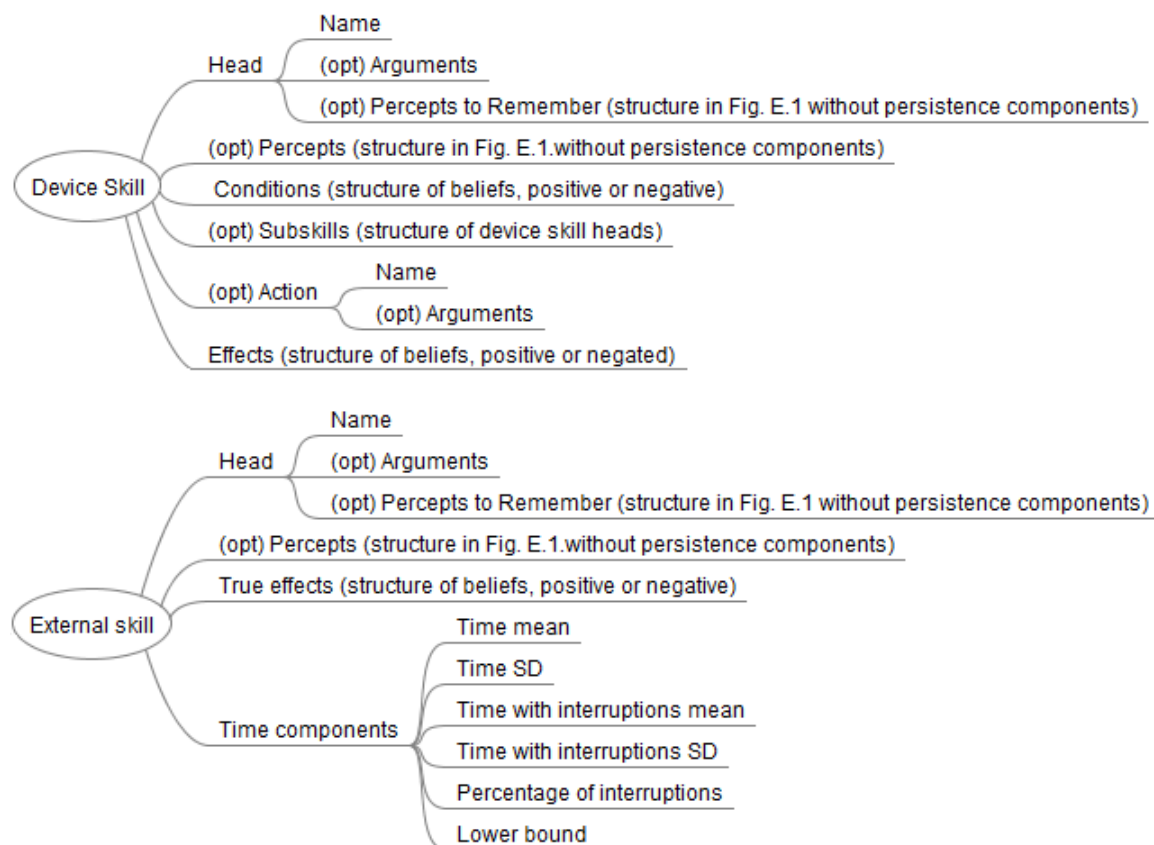


Figure E.3: Device and external skill components

describe their components separately. They are presented schematically in Fig. E.3.

E.0.1.4.1 Device Skills

Device Skills² in my modelling approach keep, and enhance, most of the components of skills as defined in ICARUS. A device skill's **head** is defined as in ICARUS by a **name** and an optional list of **arguments** presented as variables, which represent the percept names on which the device skill is applied. Additionally, I have added to the head an optional set of **percepts to remember**, which define the formats of the percepts that the simulated user should remind herself of and remember, if she perceives them once the device skill is achieved. Percepts to remember are represented as normal percepts, without the persistence components. The values of their attributes need to take a real literal value, or to have the keyword value 'any'. The modeller should only define this component for a device skill's head if the simulated user should always remember

²Please note that device skills belong to the model of the user, and not to that of the device. I have called them "device skills" to avoid a long name for them, e.g. "skills on the device"

the same type of information once the skill is achieved. More details about how they are used will be provided in the description of the remembrance/forgetfulness mechanism in the ‘Behaviour’ section.

A device skill can contain as an optional component a set of **percepts**, defining those percepts which must be present in the perceptual buffer such that the device skill is applicable. In my approach these percepts and their attributes do not contain the persistence components. As in ICARUS, the percepts which are provided here may or may not fully specify the head’s arguments (if any). In particular, I added here that they should fully specify the arguments if the device skill requires them to be visible by the simulated user (i.e. present in her perceptual buffer). For example, the device skill of monitoring a patient could require that the patient be visible on the system, and thus the argument provided as a variable, say “?p” in the device skill’s head needs to be completely defined as a patient-type percept in the device skill’s ‘percepts’ to ensure that the simulated user perceives it. Percepts should not fully specify all, or any, of the arguments when the device skill is expected to be called for achieving a goal or by a superskill with some arguments which the simulated user is intended to first find on the system (the device skill’s first subskill(s) are about finding that information). For example, a second device skill of monitoring a patient could not include a definition of the argument from the device skill’s head which refers to her, and instead include as its first subskill the finding of the patient (the notion of a subskill will be detailed below). Please note here the difference from a concept’s percepts, which must always fully specify its arguments in my approach. Each percept should only include those attributes which are needed for the simulated user’s remembrance mechanism (please see the discussion from the end of this section), as unordered. As in concepts, the values of the attributes can only be variables, which will match against the values of the attributes of the percepts from the perceptual buffer that match with the percept.

A device skill should contain as a compulsory component a set of **conditions** referencing the heads (name and arguments) of concepts which must hold/not hold if the name is negated (i.e. exist or not as instantiated beliefs in belief memory) such that the device skill is achievable. For example, a device skill of accessing a patient’s details might require that the patient (defined in the skill’s percepts) be selected on the system. Also, a device skill of updating a patient’s information might require that this information (defined as the value of the patient’s attribute, or as a separate percept in the skill’s percepts) is not updated yet (negation). In my approach, arguments must be variables which have already been defined in the device skill’s ‘percepts’, while

in ICARUS new variables with the meaning of ‘any’ can also be used. I have not implemented this convention of ICARUS in this respect, as I did not find it useful for my models.

A list of **subskills** are an optional component of a device skill. Subskills reference the heads (name, arguments, percepts to remember) of skills (device or external) which must be carried out, in their given order, for achieving the device skill. In contrast to ICARUS, the arguments provided in the subskills must be variables which have been defined in the device skill’s ‘percepts’ component (and not also in the ‘conditions’ component), and there cannot be additional variables with the meaning of ‘any’. Moreover, the values of the subskills’ **percepts to remember** need to be either provided as variables which must have been defined in the device skill’s ‘percepts’ field, to take a real literal value, or to have the keyword value ‘any’.

A device skill may include as an optional component an **action**, representing an intended action that the simulated user would want to carry out on the device for achieving the device skill. If the device skill is achievable, this action will be forwarded to the device for being carried out, as we will see in the description of the skill execution behaviour. An action is defined by a **name** and an optional set of **arguments**. As opposed to ICARUS, arguments from my approach must have been defined in the device skill’s ‘percepts’ component. In ICARUS, they can also be defined in the skill’s ‘conditions’ component, or be numerical or literal, but I have not found this useful for my models.

A device skill must always include a set of **effects**. They represent those concepts which are expected to become applicable, by being instantiated into beliefs in the belief memory, or not (if negated), and thus be absent in belief memory, once the device skill is carried out. A device skill is only achievable if the concepts from its effects do not hold/hold (if negated) (and if those from its conditions do/do not if negated, as mentioned above). As in ICARUS, effects are defined as the heads (name, which can be negated, and arguments) of the expected/not expected beliefs.

I also added a convention about the percepts which should go into the device skill’s ‘percepts’ component: only those percepts which are needed for the device skill, and to be sent as arguments to its ‘conditions’, ‘subskills’ or ‘action’. In contrast to the notion of concepts from my approach, percepts should also include all of the attributes which are checked for them in the definitions of the concepts which correspond to those from their ‘conditions’ component. They should not also include the attributes that are required by the definitions of the skills corresponding with those from ‘subskills’,

as they would be checked by these skills themselves. This decision has to do with the functionality of the simulated user reminding herself of those percepts and their attributes which are used in skills, as will be described for the remembrance mechanism in the ‘Behaviour’ section.

As for skills in ICARUS, the device skills from my modelling approach should include either a ‘subskills’ component, or an ‘action’ component, but they cannot include both. Conceptually, the simulated user can either follow certain steps (subskills) to accomplish the device skill, or directly carry out an action on the device. Device skills which include a ‘subskills’ component are non-primitive, while the ones which include an ‘action’ component are primitive, as they do not depend on any other skills. Skills thus form a hierarchy, with non-primitive skills at the top and primitive ones as the last elements at the bottom (the leaves).

Examples: Let us assume that one of the simulated user’s device skills is to access the page containing a patient’s alerts, and that the user would only perform this device skill if the patient would require monitoring (as she would not normally check the alerts of patients who don’t). To perform the skill, let us further assume that the simulated user needs to first click on a link from the patient’s second name on the homepage to go to the patient’s Demographics page, and then on an “Alerts” button from there to go to the patient’s alerts. This would be written in the following way in terms of device skills:

Head: (access_alerts ?p)

Percepts: ((patient_website ?p (highlighted ?h)) (location ?l) 1 TRUE)

Conditions: ((needs_monitoring ?p) (on_homepage ?p))

Subskills: ((go_to_demographics ?p) (go_to_alerts ?p))

Effects: (on_alerts ?p)

This device skill has the name `access_alerts` and takes one argument (a variable) defined as a percept of type “patient”, from the website, containing information and having attributes of type “highlighted” and “location”. For being achievable, the (positive) concepts from the device skill’s conditions (`needs_monitoring` and `on_homepage`) must hold for the percept by having instances in beliefs, while the one from its effects (`on_alerts`) must not. The device skill can be accomplished by following in order two subskills referencing the device skills `go_to_demographics` and `go_to_alerts`, and the expected effect is that the concept `on_alerts` will have an instance in beliefs for the

percept. Therefore, the simulated user can access the alerts for a patient by going to her Demographics page and then to her Alerts page, if this patient is on the homepage and needs monitoring. The expected effect of the device skill is to end up on the patient's Alerts page.

The following would be a definition of the device skill's subskills:

Head: (go_to_demographics ?p)

Percepts: ((patient_website ?p ((second_name ?n) (location ?l)) 1 TRUE) (link_website ?li (text ?t) 1 FALSE))

Conditions: ((on_homepage ?p) (demographics_link_for_patient (?li ?p)))

Action: (click_demographics (?p ?li))

Effects: (on_demographics ?p)

Head: (go_to_alerts ?p)

Percepts: ((patient_website ?p (location ?l) 1 TRUE) (button_website ?b (text ?t) 1 FALSE))

Conditions: ((on_demographics ?p) (alerts_button ?b))

Action: (click_alerts (?p ?b))

Effects: (on_alerts ?p)

The simulated user can therefore only go to the Demographics page of the patient if she is on the homepage and has an associated link for the Demographics. She can access the page directly by carrying out the action `click_demographics` for the patient using the link, and the expected effect is that the patient's Demographics page will be accessed. For being able to access the patient's Alerts page, the simulated user should be on the patient's Demographics page and have an available "Alerts" button. This button needs to be used this time for accessing the page, with the expected effect that the page will change into the patient's Alerts page.

For clarity, I also add the definition of the concepts which were not already defined in the previous examples:

Head: (on_homepage ?p)

Percepts: (patient_website ?p (location ?l) 1 TRUE)

Tests: (== ?l homepage)

Head: (on_demographics ?p)

Percepts: (patient website ?p (location ?l) 1 TRUE)

Tests: (== ?l Demographics)

Head: (on_alerts ?p)

Percepts: (patient website ?p (location ?l) 1 TRUE)

Tests: (== ?l Alerts)

Head: (demographics_link_for_patient ?b ?p)

Percepts: ((patient website ?p (second_name ?n) 1 TRUE) (link website ?li (text ?t) 1 FALSE))

Tests: (== ?t ?n)

Head: (alerts_button ?b)

Percepts: (button website ?b (text ?t) 1 FALSE)

Tests: (== ?t Alerts)

A patient is on any of the three pages if her “location” attribute has as value the name of the page. A Demographics link is available for the patient if its text contains her name, and an Alerts button is available for her if its text is “Alerts”.

E.0.1.4.2 External Skills

As already mentioned, **external skills** are a simplified view of a real user’s activity in her static environment, and added by me as additional to the general notion of skills from ICARUS. We assume that they are always directly achievable, with the expected effects, as they are completely in the hands of the simulated user. This means that a ‘conditions’ or ‘subskills’ component is no longer necessary for them, and that instead of ‘effects’ we could say ‘true effects’ to reflect the fact that they will always be acquired. Moreover, as they are in the simulated user’s hands, it will depend on her how long it takes for the external skill to be performed, while in the case of the user’s device skills the device would determine the duration of any action sent to it. This means that components having to do with time are necessary for external skills.

The **head** component of an external skill is defined as for device skills. The optional set of **percepts** are this time, as opposed to the same component from device

skills, the simulated user's safety check that she has perceived (i.e. has available) the needed percepts for carrying out the external skill. These percepts are defined as regular percepts, without the persistence components which are related to the simulated user's remembrance/forgetfulness mechanisms and, as opposed to the percepts from device skills, they must always define the head's arguments (if any). Moreover, the values of their attributes must be provided directly as numeric or literal values (as opposed to variables for device skills), to be checked against the values of the attributes of the matching percepts from the perceptual buffer. An external skill's **true effects** component includes those concepts which the simulated user will automatically consider are applicable, and thus instantiate as beliefs in her belief memory, as a result of the external skill. They should not reference existing concepts, as should a device skill's effects. As for a device skill's effects, they are defined as the heads (name, which can be negated, and arguments) of the expected/not expected beliefs. Please note the distinction between a device skill's 'effects', representing the simulated user's expectation about concepts being applicable or not, and an external skill's 'true effects', which the simulated user automatically considers are applicable or not.

The following are the components (which I call '**time components**') of an external skill which will help compute its duration probabilistically, where I am assuming that the time is normally distributed:

- **Time mean:** the mean expected time of the external skill
- **Time SD:** the standard deviation of its expected time
- **Time with interruptions mean:** the mean expected time of the external skill in the presence of interruptions to the simulated user
- **Time with interruptions SD:** the standard deviation of its expected time in the presence of interruptions to the simulated user
- **Percentage of interruptions:** the chance of the simulated user being interrupted in her work
- **Lower bound:** the minimum value of the external skill's duration, such that this duration is not unrealistic as generated randomly, in the presence of interruptions or not

Example: A possible external activity that a real user could perform in relation to a telehealthcare system is to write down her outcome on the patient state on paper,

either for statistic purposes in her team or for recording them in the patient's paper record. Let's suppose that she would only normally do this when she is perceiving the patient from an "Alerts" page from the website, where she has entered the patient's outcome. An external skill for the activity would thus need to check that the current page is the Alerts page, and that the patient whom she perceives is also placed there. This would be represented as percepts within its component with the same name. Moreover, let's suppose that the writing down of the information could take a user (real or hypothetical) at least 5 seconds, a mean of 8 seconds and a standard deviation of 1 second under normal work conditions, and if interrupted a mean of 12 seconds and a standard deviation of 3 seconds. This information would therefore be defined in the external skill's time components. Moreover, let's say that the simulated user would instantiate a concept named "outcome_written_down" and add it to her belief memory once the external skill is performed. We could represent this external skill thus (I am considering the time in seconds here, although for greater expressiveness milliseconds could be used):

Head: (write_down_outcome ?p)

Percepts: ((patient_website ?p (location Alerts) 1 TRUE) (page_website ?pag (name Alerts) 1 FALSE))

True Effects: (outcome_written_down ?p)

Time Components: (8 1 12 3 5)

E.0.1.5 The Intention Memory

The intention memory stores, as in ICARUS, **intentions**. An intention is an instance of a *skill* showing at any one time what the next activity that the simulated user wants to pursue is, and by using which percepts from the simulated user's perceptual buffer. In our case, intentions can instantiate both device and external skills. An intention instantiates a device skill by attributing percepts from the perceptual buffer (their names and values of their attributes) to the device skill's 'percepts' component, and their names as arguments to the device skill's 'conditions', 'subskills'/'action' and 'effects'. The 'percepts to remember' component of a device skill's head or of its 'subskills' component has a more complicated mapping with percepts, and I will return to it in the section on remembrance/forgetfulness mechanisms. An intention instantiates an external skill by attributing the names of the percepts from the perceptual buffer to the

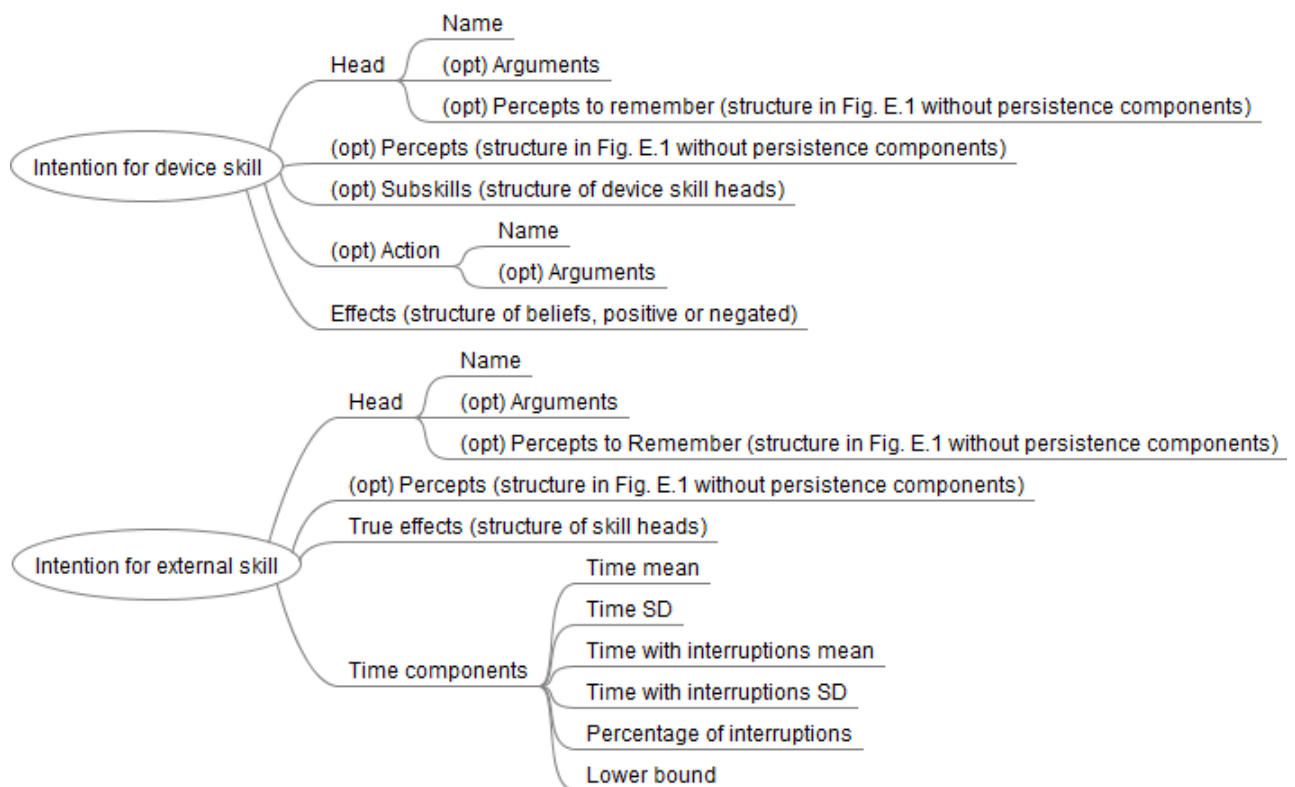


Figure E.4: Components of intentions instantiating device skills and of those instantiating external skills

external skill's 'percepts' component (which had their attributes' values already defined), and as arguments to the external skill's 'true effects'. The structure of intentions, as distinguishing intentions instantiating device skills and intentions instantiating external skills, is presented in Fig. E.4.

Intentions in my modelling approach have the same components as skills, with the exception (unlike in ICARUS) that I have not included a 'conditions' component for intentions for device skills, as conditions are checked before instantiating the device skill into an intention, as will be explained in the 'Behaviour' section.

Examples: The following would be the intentions and subintentions derived as instances of the skills and subskills defined in the examples from above if they are found to apply for perceived percepts p1, li1 for the link to the Demographics page from the patient's surname, and b1 for the button for the Alerts page from the Demographics page. They all have the arguments replaced by the names of the real percepts, and the values of their percepts' attributes as the values of the real percepts' attributes.

Head: (access_alerts p1)

Percepts: ((patient_website p1 (highlighted_red) (location_homepage)) 1 TRUE)

Conditions: ((needs_monitoring p1) (on_homepage p1))

Subskills: ((go_to_demographics p1) (go_to_alerts p1))

Effects: (on_alerts p1)

Head: (go_to_demographics p1)

Percepts: ((patient_website p1 ((second_name Doe) (location_homepage)) 1 TRUE) (link_website li1 (text Doe) 1 FALSE))

Conditions: ((on_homepage p1) (demographics_link_for_patient li1 p1))

Action: (click_demographics p1 li1)

Effects: (on_demographics p1)

Head: (go_to_alerts p1)

Percepts: ((patient_website p1 (location_Demographics) 1 TRUE) (button_website b1 (text Alerts) 1 FALSE))

Conditions: ((on_demographics p1) (alerts_button b1))

Action: (click_alerts p1 b1)

Effects: (on_alerts p1)

E.0.1.6 Goals

As I will explain in the ‘Behaviour’ section, I have made the decision not to implement the functionality of problem solving of ICARUS in my modelling approach. Nevertheless, I have decided to use its notion of goals, as additional to STM and LTM constructs. Goals represent, as in ICARUS, what the simulated user would like to achieve or, more exactly, what she would like to come to believe. They are the reason behind skill execution. As opposed to ICARUS, where goals were organised in a set and could be accomplished in any order, in my approach they are organised in a list, which the simulated user will attempt to accomplish in order. Please be reminded that the user state from my initial modelling approach was defined by a list of goals and a belief state (as was presented in Chapter 5). To keep our current approach consistent with this, and having the fact that we also have percepts and intentions with which the simulated user works with at any one time here, we could consider that the list of goals and STM

define the simulated user's state at any one time.

As in ICARUS, goals in my approach have a **name** and a list of **arguments**. Also as in ICARUS, goals can also be positive or negated, to represent what the simulated user would like/not like to believe any longer. In contrast to ICARUS though, goals in my approach need to be given arguments representing real percept names, and they cannot also be given variables with the meaning of 'any'. Instead, goals with one argument (only) can have it defined as the keyword literal 'any' or 'all', in which case they are named **repeatable goals**. The modeller will need to define for each of the names of such goals where they should be taking their real argument values from. During the simulation run, the repeatable goal will be transformed into multiple goals, each taking one of the possible values for the argument.

Together with the decision of not implementing problem solving, I have added the convention that **goals in my modelling approach should always be directly achievable by a skill**. I will explain the functionality of achieving goals in the 'Behaviour' section.

Example: An example of a user goal would be that of reaching a state where the user is on the patient's, let's say "p1", Alerts page. The goal would be represented as:

on_alerts p1

The name of the goal is therefore "on_alerts" and its argument is "p1". Please note that I have chosen this goal such that its name corresponds to the effects of the device skill *access_alerts*. This correspondence would make the device skill be chosen for attempting to achieve the goal, as will become clear from the 'Behaviour' section.

E.0.2 Notions for the Device Model

As I already mentioned, I am still representing the device as a basic LTS. At any one time, the device is in a certain **state**, and the simulated user can require the accomplishment of an action from its **set of actions** which, if available and possible for that state, will lead by means of a **transition relation** to a different device state.

E.0.2.1 The Device State

I define the device state as consisting of a set of **device state components**, representing the elements of the device which can be visible or not by the simulated user. I have

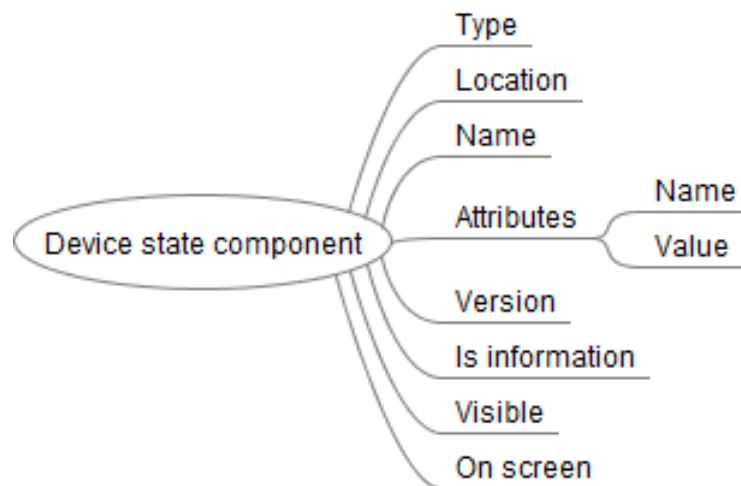


Figure E.5: Device state components

represented device state components at the same level of granularity and very similarly to percepts. In particular, they contain the same components as percepts, with the same representation and meaning, apart from the persistence components for themselves or their attributes (Fig. E.5). Moreover, I have added two additional components to them:

- **On screen:** a truth value (TRUE/FALSE) indicating whether the device state component has been rendered by the device on its screen, or it is internal to the system (e.g. related to its database)
- **Visible:** a truth value (TRUE/FALSE) indicating whether the device state component is being displayed for the simulated user within the margins of her display. Please note that, although being rendered, a device state component may not be visible to the simulated user if it is hidden outside the margins of her display and she would need to scroll for it.

I should add here that the modeller may decide to impose the use of some pre-defined types for device state components. For example, for structural-type device state components (i.e. those with ‘is information’ equal to FALSE), she may decide that appropriate types could be “button”, “menu”, “page”, “table”, etc. For information-type device state components (i.e. those with ‘is information’ equal to TRUE), she may decide that they are ‘patient’, “alert”, “measurement”, etc. Moreover, she may impose different possible ‘location’ components, e.g. “website”, “spreadsheet”, “OS”.

Example: Let’s assume that the simulated user is on one of the pages of the telemonitoring website, where she can see at most 10 rows in a list with patients

without the need to scroll, but that she also works with a spreadsheet (as in my study from Lothian) which is open on a different tab. I assume that her screen resolution is 1024x768, so that any information which has $x\text{-end} < 1024$ and $y\text{-end} < 768$ would be visible by the simulated user. The following are some device state components which could appear in this case. The first is completely visible to her, the second is on screen but not visible, and the third is not on screen (or visible):

(patient website p1 ((name Helen) (x-start-position 20) (y-start-position 300)(x-end-position 420) (y-end-position 320) 1 TRUE TRUE TRUE)

(patient website p11 ((name Jack) (x-start-position 20) (y-start-position 800)(x-end-position 420) (y-end-position 820)) 1 TRUE TRUE FALSE)

(textarea spreadsheet t1 (text "Please enter your comments here") 1 FALSE FALSE FALSE)

All apart from the last two components of these device state components should be familiar to the reader from percepts, and the last two components represent, in order, the ‘on screen’ and ‘visible’ components. For the first two examples from above, it is clear that the ‘on screen’ and ‘visible’ components have been set up according to the coordinates of the information. If such a level of detail is required, it needs to be implemented by the modeller for the population of information-type device state components (an input for the device model; this has been explained in Chapter 7).

E.0.2.2 The Device Actions

The set of device actions includes all of the possible actions that the simulated user could carry out on the device. A device action is conceptually different from the simulated user’s action from an intention instantiating a primitive device skill (which I call here ‘user action’). A user action is only intended by the user, but may not have a corresponding device action (i.e. it may not exist for the device) or, even if it does, this corresponding action may not be possible for that device state. As I will describe in the ‘Behaviour’ section, I have implemented functionality which can identify such mode confusion problems and return a warning to the modeller.

The structure of device actions is provided in Fig. E.6.

A device action contains as components a **name**, an optional list of **device state**

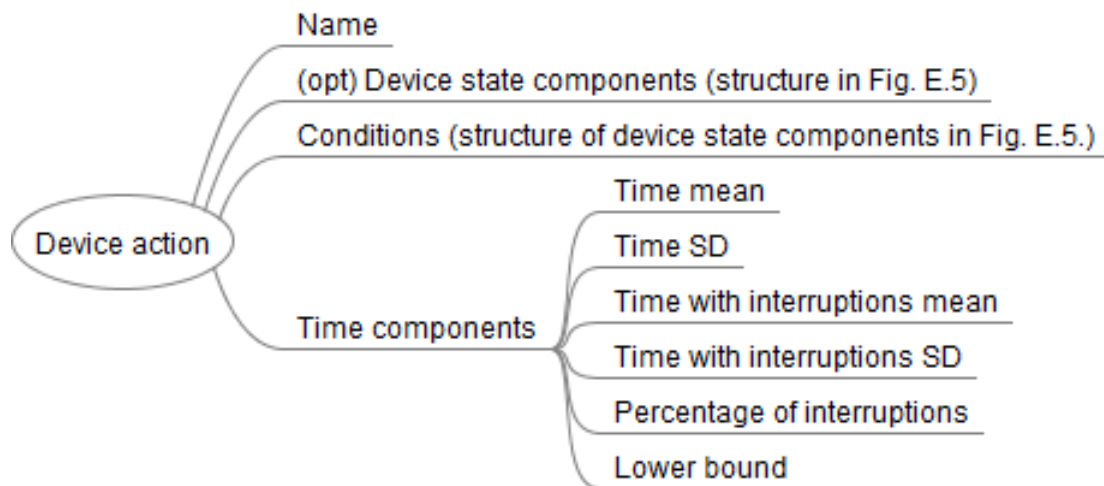


Figure E.6: Device action components

components, a set of **conditions**, and **time components** which help compute the duration of the device action. Its **device state components** define the arguments that the device action could take. Their names and attribute values must be provided as variables, and they should match against those of the percepts which are bound to the arguments from the head of a user action³, and in this form also be present on the device state. I have not used the notion of a head containing the name and arguments as variables for the device action, as these arguments would not be used in other components of the device action. An action's **conditions** are also represented as device state components, but this time they have real values for their attributes. They represent device state components, and with required attribute values, which should be present on the device such that the device action is possible. I have considered that the device action's duration includes user movement time, because of difficulties in collecting data on user movement time, as was described in Chapter 8. The following are the components which will help compute the duration of the device action probabilistically, where we assume that the time is normally distributed:

- **Time mean:** the mean expected time of the device action
- **Time SD:** the standard deviation of its expected time
- **Time with interruptions mean:** the mean expected time of the device action in the presence of interruptions to the simulated user

³Please note here that percepts are expected to match with device state components. This is because percepts are in fact perceived device state components, to which the simulated user adds persistence components needed for remembrance/forgetfulness. More details will be provided in the section on perception.

- **Time with interruptions SD:** the standard deviation of its expected time in the presence of interruptions to the simulated user
- **Percentage of interruptions:** the chance of the simulated user being interrupted while performing the action on the device
- **Lower bound:** the minimum value of the device action's duration, such that this duration is not unrealistic as generated randomly, in the presence of interruptions or not

I propose as future work the separate consideration of user movement time, as was discussed in Chapter 10.

Examples: I have mentioned in the examples for device skills the user actions named “click_demographics” and “click_alerts”, taking one argument referencing a percept of type “patient”. The following could be associated device actions for these user actions, where I have considered seconds for the unit of time:

Name: click_demographics

Device state components: ((patient website ?p ((second_name ?n) (location ?l) 1 TRUE TRUE TRUE) (link website ?li (text ?t) 1 FALSE TRUE TRUE))

Conditions: ((page website ?p (name homepage) 1 FALSE TRUE TRUE))

Time components: (2 0.5 4 1 1.5)

This device action can therefore be called with percepts which would be equivalent to device state components with type “percept” and “link”, and having certain attributes. It is only possible if the device contains such equivalent device state components, and additionally is in a state where the page is the homepage, i.e. there is a device state component with type “page”, and an attribute “name” with the value “homepage”, which is structural, and both visible and on screen. Its duration is defined by a time mean of 2 s, time standard deviation of 0.5 s, time with interruptions mean of 4 s, time with interruptions standard deviation of 1 s and lower bound of 1.5 s.

Name: click_alerts

Device state components: ((patient website ?p (location ?l) TRUE TRUE TRUE) (button website ?b (text ?t) 1 FALSE TRUE TRUE))

Conditions: ((page website ?p (name Demographics) 1 FALSE TRUE TRUE))

Time components: ((2 0.5 4 1 1.5)

E.0.3 Behaviour of the User Model

E.0.3.1 Perception

Perception is the first step performed by the simulated user at the beginning of a cognitive cycle. I have kept it as very simplistic for the purposes of my examples, the simulated user always seeing (i.e. perceiving) all of the device state components which have been rendered on screen by the device (i.e. have 'on screen'=TRUE) and are visible at that time within the display margins (i.e. have 'visible'=TRUE).

The following is a high level description of the steps performed by the simulated user for perception, in which I do not include remembrance steps (i.e. what happens with persistence components), which will be described in the section on remembrance/forgetfulness:

1. Create an associated percept for the device state component (which I will call perc1), by using the common components of these two notions- type, location, name, attributes, version, is information.
2. Retrieve any other percept from the perceptual buffer (which I will call perc2) which is equivalent with the newly created percept. I define a percept as 'equivalent' to another if their name-version pairs are equal.
3. If an equivalent percept perc2 is found, there are two possibilities:
 - (a) It is an old percept which was remembered by the simulated user. In this case, we replace perc2 with the new percept perc1, making sure that perc1 inherits the status in memory of perc2, and mark perc1 as a newly perceived percept. More details on these steps will be provided in the section on remembrance/forgetfulness.
 - (b) It is a new percept, already perceived on this cognitive cycle. As it is not allowed for equivalent percepts to coexist in the perceptual buffer, this will issue an error message and the run will be terminated.
4. If no percept perc2 is found, percept perc1 can be added to the perceptual buffer as a new percept.

The use of a very simple, all observing, perception functionality means, of course, that I have not considered cases where the user would only be looking for certain information on the device, neglecting the rest, or issues such as eye movement or

attention. Moreover, I am not attributing time to the simulated user's perception (instead, I use external actions to add it wherever I observed real users spending some time in reading the information, as I will explain in future sections). My perception functionality is nevertheless separate within the implementation of the simulated user's behaviour, and it can be changed with a more psychologically correct one. I propose as future work the collaboration with a psychologist for this purpose.

E.0.3.2 Conceptual Inference

Like in ICARUS, conceptual inference is one of the most important functionalities in my modelling approach. It represents the process of instantiating generic concepts from the LTM conceptual memory into beliefs in the STM belief memory, by using percepts which match from the simulated user's current perceptual buffer.

The conceptual inference process starts at the beginning of a cognitive cycle, once the simulated user has finished her perception, and the resulting update of the percepts from her perceptual buffer. Given the simulated user's (fixed) set of concepts from conceptual memory, and the current perceptual buffer:

1. The simulated user picks a concept.
2. She matches the concept's percepts from the component with the same name with her real percepts from the perceptual buffer. Two percepts, one from the concept's percepts and one from the perceptual buffer, 'match' if they have the same 'type', 'location', 'is information' components, the same types of attributes and no additional ones. In this case, the names and attribute values of the real percepts will be considered to replace the variables of the concept's percepts for the rest of the concept's components. Combinations of all of the possible matching percepts are considered for the set of percepts of the concept, each constituting what is called a potential 'binding'. Illegal bindings, those where the same percept is matched several times with different percepts from the concept's percepts, are dismissed.
3. For each of the obtained bindings:
 - (a) If the concept is non-primitive (i.e. has a 'relations' component), the simulated user checks whether its relations hold. A concept's relations hold if the beliefs required by them exist/do not exist if negated, in her belief memory. To this end, the simulated user instantiates each belief/negated

belief from her concept's 'relations' component with the potential binding by replacing arguments with the real percept names, as they were bound for 'percepts'. Then, for positive beliefs, she checks whether the same belief exists in her belief memory, and for negative ones she checks whether it does not exist there. A negative result for any of the beliefs indicates that the chosen binding is not possible for the concept, and the simulated user will try out the next binding.

- (b) If all of the beliefs from 'relations' hold or not as required, or the concept is primitive, the simulated user checks whether the arithmetic or logical tests required by the concept's 'tests' field are respected. To this end, she instantiates the attribute values included in the tests with the real values of the percept attributes from the binding, and checks the resulting relation. If the result is FALSE, then the chosen binding is not possible for the concept and the simulated user will try out the next binding.
- (c) If the 'tests' also hold, then the concept is applicable. The simulated user replaces the arguments from the concept's head with the names of the real percepts from the binding to create a belief, and, if such a belief does not already exist in the belief memory, she adds it there. Once a belief is added to belief memory, higher-level concepts from the concept hierarchy may also become applicable, be converted into beliefs and added there as well. The simulated user then starts again, with a new concept.

4. If there is no matching binding for the percepts, or none of the bindings led to applicable 'relations' or 'tests', then the concept is not applicable, and the simulated user will pass on to trying out the next concept.

This process continues until there are no more new beliefs generated from the set of concepts. To sum up, a primitive concept is applicable when its elements from 'percepts' match real percepts from the perceptual buffer and when its relations from 'tests' are TRUE when variables are replaced with the matching percepts' attribute values. A non-primitive concept is applicable when the same apply, and additionally when the beliefs/negated beliefs from its 'relations', with variables replaced by percept names from the binding, exist or not (if negated) in the belief memory. I must also emphasise on the importance of replacing the same variables with values consistently throughout the concept: if, say, variable ?p1 appears in the concept's head's arguments, percepts

and in relations, then it must be bound with the same value taken from one percept which matches it.

Example: Let us consider the following concept provided in previous examples:

Head: (needs_monitoring ?p)

Percepts: (patient website ?p (highlighted ?h) 1 TRUE)

Tests: (== ?h red)

Also, let's assume that the simulated user has the following percepts in her perceptual buffer (persistence components were omitted):

(patient website p1 ((name Helen) (condition COPD) (highlighted red)) 1 TRUE)

(patient website p2 ((condition COPD) (highlighted red) 1 TRUE)

Both of these percepts will match with the concept and make it applicable. This is because they have the 'type' (patient), 'location' (website) and 'is information' (TRUE) components required by the concept's percepts, and both of them contain an attribute of type "highlighted". Moreover, if we replace in the concept's tests the variable with the value of the 'highlighted' attribute for each of them, the relation is true. This will result in the concept being instantiated into two beliefs: *needs_monitoring p1* and *needs_monitoring p2*.

Let's further consider the following concept:

Head: (COPD_patients_needing_monitoring (?p1 ?p2))

Percepts: ((patient website ?p1 (condition ?c1) 1 TRUE) (patient website ?p2 (condition ?c2) 1 TRUE))

Relations: ((needs_monitoring ?p1) (needs_monitoring ?p2))

Tests: ((== ?c1 ?c2) && (== ?c1 COPD))

The same percepts will match at the same time with it out of similar reasons as above, and also because the beliefs from 'relations', when instantiated given the binding into "needs_monitoring p1" and "needs_monitoring p2" were deduced from the first concept. This results into a new belief: *COPD_patients_needing_monitoring p1 p2*.

Please note that, if there are several percepts respecting the requirements of the

concept, this could lead to a multitude of beliefs regarding their relationship. For example, if we had the following 3 additional percepts:

(patient website p3 ((name John) (condition COPD) (highlighted red)) 1 TRUE)

(patient website p4 ((surname Robertson) (condition COPD) (highlighted red) 1 TRUE)

(patient website p5 ((condition CHF) (highlighted red) 1 TRUE)

Then, apart from the existing beliefs from above, the following beliefs would be generated: *(needs_monitoring p3)*, *(needs_monitoring p4)*, *(needs_monitoring p5)*, *COPD_patients_needing_monitoring p3 p4*, *COPD_patients_needing_monitoring p1 p3*, *COPD_patients_needing_monitoring p1 p4*, *COPD_patients_needing_monitoring p2 p3*, *COPD_patients_needing_monitoring p2 p4*. Please note that the last percept has only matched the first concept, as the value of its ‘condition’ attribute does respect the ‘tests’ of the second one.

E.0.3.3 Skill Execution

Skill execution represents the main mechanism by which the simulated user chooses and attempts to accomplish a skill. As in ICARUS, in my approach the simulated user will launch skill execution if she is directly required to accomplish a certain skill. Alternatively, it will be invoked by her in an attempt to achieve a list of goals, if such a list has been provided as input to the user model. The latter is the main way in which users of a telehealthcare system would be simulated in doing their work, and therefore I will focus on it in this description.

I have decided not to implement the problem solving behaviour of ICARUS in my modelling approach. This is due to the fact that the users of the telehealthcare system that I have assessed in my usability study in Lothian, even if beginners in the project, were experienced enough with their routine work on the system, which is what I wanted to model. Even if slower in their actions, I have never witnessed them not knowing what to do on the system and needing to look for a solution.

I have kept from the problem solving behaviour of ICARUS the notion of goals, adding to it the fact that goals should always be directly achievable by a skill (otherwise the simulation run returns an error message). Put another way, the simulated user should always know what to do to accomplish her goals, which reflects my observations of real

users of the telehealthcare system from Lothian. Moreover, the goals in my approach are organised in a list and are addressed in order by the simulated user.

The simulated user will go through the following steps for processing goals through skill execution:

1. Pick the next goal in her list of goals.
2. Check if the goal was not already achieved. A positive goal is considered as achieved when an associated belief (i.e. with the same name and arguments) exists in belief memory. A negated goal is considered as achieved when, once transformed into positive (i.e. by removing its negation), an associated belief does not exist for it in belief memory.
3. If the goal was already achieved, remove it from the list of goals and pass on to the next goal (return to point 1).
4. If the goal was not already achieved, look for skills which would lead to its achievement and make a list of these potential skills. A device/external skill achieves a goal if it has a belief with the name of the goal in its effects/true effects.
5. If no skills are available to achieve the goal, return an error message and terminate the whole run.
6. Use a heuristic (as in ICARUS for problem solving) to identify the best skills among the list of potential skills to be chosen for achieving the goal, and reorganise the list of potential skills such that they occupy the first positions. The heuristic that I implemented chooses those skills which would accomplish the highest number of other goals from the list of goals, but other heuristics could also be used.

Please note that in ICARUS a heuristic was implemented which identifies those skills which would have most of their conditions satisfied. This would only apply in our case for device skills, which can have conditions. A device skill's conditions are considered as satisfied if each of their positive beliefs have an associated equal belief from the belief memory, and each of their negated beliefs, once transformed into positive (i.e. their negation is removed), do not have an associated equal belief from belief memory. This heuristic is only useful in

conjunction with problem solving, as it implies that device skills which do not have all of their conditions as satisfied could still be accomplished.

7. For each of the skills in the list of potential skills, taken in order, the simulated user will:
 - 7.1. Match the skill's head's arguments with the arguments required by the goal.
 - 7.2. If the skill has percepts, match them with real percepts from the perceptual buffer to create potential bindings. This is done the same way as it is done for a concept's percepts during conceptual inference, with the difference that those percepts which define the arguments in the skill's head should be matched with real percepts having the name required by the arguments from the goal.
 - 7.3. If matching percepts are not found for the skill's percepts, go back to trying out another skill from the list of potential skills.
 - 7.4. For each potential binding, the simulated user will:
 - 7.4.1. Check that the skill given the binding was not already achieved. A device/external skill is considered as achieved for a binding when each of the positive beliefs from its effects/true effects have an associated equal belief in belief memory, and when each of its negated beliefs from effects/true effects, once transformed into positive (i.e. their negation is removed), do not have an associated belief from belief memory. The simulated user needs to first replace the arguments from the effects'/true effects' beliefs with the names of the real percepts from the binding, and then make this check.
 - 7.4.2. If the skill was already achieved for the binding, return a warning message and pass on to achieving the next skill in the hierarchy of skills or the next goal. This cannot happen if the skill is chosen to directly address a goal, as the achievement of the goal was already checked in point 2, but only if it is chosen to address a subskill of another skill.
 - 7.4.3. If the skill was not already achieved for the binding and it is a device skill, check that the skill's conditions are satisfied. A device skill's conditions are considered as satisfied if each of their positive beliefs has an associated equal belief from the belief memory, and each of their negated beliefs, once transformed into positive (by having the negation

removed), do not have an associated equal belief from belief memory. As for effects, the simulated user needs to replace the arguments of the conditions' beliefs with the names of real percepts from the binding, and perform the check.

- 7.4.4. If the skill is a device skill, and one or more of its conditions are not satisfied by any of the bindings, return a warning message and pass on to the next skill in the list of potential skills.
- 7.4.5. If the skill is a device skill and all of its conditions are also satisfied, or it is an external skill, the simulated user will consider the skill as achievable and do the following:
 - 7.4.5.1. Mark the information-type percepts and their information-type attributes that match from the binding as having been remembered and used by the skill. I will return to this point in the description of the remembrance/forgetfulness mechanism.
 - 7.4.5.2. Create an intention as an instance of the skill, where the skill's percepts are replaced by the real matching percepts, and the arguments from the rest of the skill's components are consistently replaced with the names of the matching percepts. As the intention will contain the full definition of the matching percepts, it is also set to keep track of their attributes which were actually required by the skill, for remembrance/forgetfulness purposes, as will be explained in the next section.
 - 7.4.5.3. If the skill is a device skill which has subskills (in its 'relations' component):
 - 7.4.5.4. Replace the arguments of the subskills with the names of the real matched percepts from the binding, and add the subskills to the component with the same name from the created intention.
 - 7.4.5.5. If the subskill has percepts to remember, replace the values of their attributes with attribute values from the binding. More details about this step will be described in the section on percepts to remember.
 - 7.4.5.6. Take each subskill in turn and attempt to accomplish it by returning to point 7.1 with its name and the binding (the latter such that the correct arguments are used).
 - 7.4.5.7. If instead the skill is an external skill:

- 7.4.5.8. Compute its time as a random number generated by the probability distribution given either by its time mean/standard deviation or time with interruptions mean/standard deviation, depending on the percentage of interruptions, and truncating the results to the lower bound. At this point, the Java application implementing the modelling approach, presented in Chapter 7, will count this time within the user's total time.
- 7.4.5.9. Reset the persistence components of the percepts which were used as part of the intention, to indicate that they are no longer in use or required to be remembered. More details will be provided in the next section.
- 7.4.5.10. Add all of the beliefs included in the external skill's 'true effects' to the belief memory, as the simulated user will always believe that the external skill is achievable.
- 7.4.5.11. Perform a new conceptual inference, as the new beliefs may make higher-level concepts from the concept hierarchy to hold.
- 7.4.5.12. Consider the intention as achieved, and pass on to accomplishing the next intention in the hierarchy, skill or the next goal if all of the intentions have been achieved.
- 7.4.5.13. If instead the skill is a device skill and has an action (it is a primitive skill), replace the arguments of the action with the names of the real matched percepts from the binding and send the resulting action to the device.

After an action is sent to the device, the device will change its state (as will be explained in the section on the behaviour of the device model). At this point, the Java application implementing the modelling approach, presented in Chapter 7, will add the device action's waiting time to the user's total time. The changing of the device state leads to a new cognitive cycle for the simulated user who will:

1. Perceive the new device state, as was described in the 'Perception' section.
2. Forget those percepts or their attributes which have exceeded a certain maximum remembrance time, or if there are too many attributes. More details about this step will be presented in the next section.
3. Clear the beliefs from the belief memory and perform a new conceptual inference.

4. Check that the latest intention's effects have been met. An intention's effects are met if all of their positive beliefs have an associated belief in belief memory, and all of their negated ones, when transformed into positive (i.e. by having their negation removed), do not have an associated belief there. This step can be removed for modelling real experienced users who do not double check the results of their work on the system.
5. If any of the intention's effects have not been met, return a warning message and terminate the simulation run.
6. Reset the persistence components of the percepts which were used as part of the intention, as will be described in the next section.
7. Consider the intention as achieved and, if the intention hierarchy does not have any intentions left, consider the goal as achieved as well. Pass on to the next intention in the hierarchy, skill or goal.

Whenever an intention is achieved, the simulated user removes it from the hierarchy of intentions, and from the list of subintentions of its parents. The achievement of an intention can also lead to the achievement of other intentions from the same level in the hierarchy, or to the achievement of intentions from higher in the hierarchy if the intention was its parent's only subintention left. Each of these intentions will also be removed from the hierarchy, and the persistence components of the real percepts which were used for them will be reset. For each of the achieved intentions or goals, the Java application implementing the modelling approach, which was presented in Chapter 7, keeps track of the time that they have taken.

E.0.3.4 Mechanisms for Remembrance/Forgetfulness

The COs who use the telehealthcare system which was described in Chapter 3 need to move information between several applications (which I considered make up the system), relying more or less on their memory. I have therefore decided that it is important to consider adding remembrance and forgetfulness mechanisms for the simulated user, which were not also implemented in ICARUS. I have developed my own, simplistic yet reasonable, such mechanisms. They have a very limited psychological basis, and therefore I propose as future work the collaboration with a psychologist for validating, evaluating them extensively against real human behaviour, and improving them.

The main premise underlying my remembrance/forgetfulness mechanisms is that humans have **limited STM, both with respect to duration and capacity**. In terms of duration, Atkinson and Shiffrin found that STM lasts for **15-30 seconds** ([37]). In terms of capacity, **the magic number 7 +/- 2 by Miller** ([157]), although largely contested, is still the most frequently cited in the literature. It suggests that a human can remember in STM as much as 7 +/- 2 items at once.

Not to constrain the modeller in her choice of values for STM duration and capacity, I decided to offer the option for her to provide them as inputs to the user model. In this way, not only can the modeller configure the simulated user with values from her choice of psychological study, but also be able to vary the memory of the simulated user for different simulations to model humans with worse or better memory, and explore how this affects efficiency.

But what represents an *item* in STM, to be counted when considering memory capacity? The obvious reply for my model of the simulated user is the percept, i.e. something which is seen will be remembered. Nevertheless, if we look more carefully, a percept could have an unbounded list of attributes, each containing information about it. **I have therefore chosen the attribute as a more realistic memory item.** Attributes can consist of words or numbers. Some studies in psychology have shown that people can 'chunk' more or less information in one memory item depending on its spoken duration ([38]), sound similarity ([70]), familiarity ([184]) and meaning ([183, 50]). I therefore propose as future work collaboration with a psychologist for refining the granularity of memory items even further.

Another observation has to do with the type of attribute that the simulated user should remember. As already mentioned in the description for percepts from my approach, their 'is information' component indicates whether they represent dynamic information which is populated on the device (e.g. if their 'type' component is "patient", "reading", "alert", etc.), or static information about the structure of the device (e.g. if their 'type' component is "page", "table", "form", etc.). This also extends to a percept's attributes, which provide the details of the information or structure. I made the realistic assumption that humans are more interested in remembering information from the system, as opposed to the system's structure, and therefore in my modelling approach **the simulated user can only remember/forget information-type percepts and their attributes**. The only exception here is for cases in which there is a need to model the relationship between percepts (i.e. through concepts) in terms of their position, and additional attributes need to be added to information-type percepts for this purpose (e.g.

their coordinates on the screen). In such a case, these attributes will not be considered as memory items. I propose as future work the collaboration with a psychologist and empirical studies to determine which types of information a real user would remember from a system, their number and duration.

STM also contains in my modelling approach the belief memory. Should beliefs also be remembered by the simulated user, or should the simulated user always re-infer these beliefs through conceptual inference? The answer is not very clear, as I expect it to depend on how important the beliefs are for the human, and it would, again, require the work of a psychologist. Remembering all of the beliefs could lead to contradictory beliefs in the belief memory, and it is also not clear how a human would react to knowing contradictory information (i.e. would she become confused, would she deal with this by relying on the latest remembered belief, or would she rely on an older one as being generated from more reliable sources?). I expect this to depend on the human's experience with the system, with computers in general and with her work. To keep things simple, **I have therefore decided that the simulated user will always re-infer beliefs, and not remember them.** As old and new equivalent percepts cannot co-exist in the perceptual buffer at any one time (only the most recently observed percept is kept), this eliminates the chance of contradicting beliefs.

In conclusion, in my modelling approach the simulated user will remember at most a certain number of information-type percept attributes, for a certain time, and she will always re-infer beliefs and not also remember them. The following are other questions which have arisen, and conventions which I have taken for the remembrance/forgetfulness mechanisms, which should all be validated in collaboration with a psychologist as future work:

- Does a human forget information before or after perceiving how the system has changed in response to her actions? I believe that the reply to this question is not clear, as other factors (which I have not considered for the simulated user), such as fatigue or attention, could intervene. I have therefore made a reasonable convention for the simulated user: that she will always forget old information (i.e. attributes of information-type percepts) after having perceived new information (i.e. new percepts) from the system, as the new information may lead her to forgetting old one.
- Does a human refresh her memory of perceived information when using this information for a skill in the same way as if she observed the information again?

If so, will she do this when checking that she can use the information for the skill, or when she can actually accomplish the skill by using the information? I have assumed in this case the latter, and my simulated user will only remind herself information which makes a skill achievable. If the skill requires in its ‘percepts’ information-type percepts with attributes, then the simulated user will remind herself of these attributes, and also of their percepts. If the skill requires information-type percepts without attributes, the simulated user will remind herself of the existence of these percepts only.

- Does a human refresh her memory of perceived information when using it for attempting concepts? I have initially considered that this is the case, and the simulated user was reminding herself of information whenever it would lead to the achievement of a concept. This information was made of, as above, information-type percept attributes and their underlying percepts, or only the percepts, if the concept required them without their attributes. Nevertheless, I later found that this mechanism was leading to the repeated inference of some beliefs, even if they were not needed any longer. I have therefore decided that the simulated user will only remind herself of information when it makes a skill, and not also a concept, achievable.
- Is a human equally likely to forget information that she perceives and uses for skills, and that which she only perceives? One would expect that the information that the human uses would be harder to forget (but still possible if it was required by a skill started early in a long-winded process). To this end, I have decided to mark the information (i.e. information-type percepts and their attributes) that is used for a skill not only as being reminded, but also set as ‘in use’, and give the simulated user a longer memory duration for such information compared to information that she perceives but does not use. The duration for the STM can therefore be defined by the modeller as two input values: one for information-type percepts and their attributes which are in use (the higher one), and one for the ones that are not. Also, as the same information could be used repeatedly within the hierarchy of skills and subskills, I have decided to have its ‘in use’ component as a number and not truth value. It is initially set to 0, and incremented once it is used by each skill/subskill, with the meaning that it was set as required by that skill/subskill. Information which has the ‘in use’ component greater than 0 should therefore be seen as in use, and have the (higher) maximum time for such

information as applied to it.

- Is a human equally likely to forget information that she knows she would need for a later skill, and that which she just perceives? One would expect that, if she would like to remember certain information for a certain later skill, then she would also remember it for longer than other information. To deal with this case, the simulated user can set in the skills' 'percepts to remember' the formats of the percepts that she would like to mark as in use if they are perceived when the skill is achieved. If percepts to match these formats are found during perception, their setting as in use will make them harder to forget than other percepts, which will improve their chances of being remembered until they become useful for a skill. I assumed that information which is needed for being used for a later skill would be remembered the same time as information which is currently being used for a skill.
- Does a human forget information once she does not need it any more for a skill? One would expect at that point for it to be at least forgotten more rapidly. This is the view that I have taken. If information which is in use is not forgotten (using the appropriate maximum time) by the time that the simulated user accomplishes the skill that had set it up this way, once the skill is accomplished it will become no longer in use for that skill. This does not mean that it is no longer in use at all, as its 'in use' component could still be greater than 0 if the information had also been required by skills from higher up in the hierarchy. If this is the case, the (higher) time for information in use would still be used for its forgetfulness. It is when all of the skills having required it have been accomplished that 'in use' will become 0 for the information. At this point, it will no longer be considered in use and the maximum time for regular information, the lower value, will be applied for its forgetfulness. Thus, at this point the information will be forgotten quicker as the simulated user no longer needs it. To sum up, while information is required by skills, it will be harder to forget by the simulated user. If it is still remembered once all of the skills having required it have been accomplished, its rate of forgetfulness will increase as the simulated user no longer needs it.

E.0.3.4.1 The Main Mechanisms

The easiest way to understand the mechanism of remembrance/forgetfulness is to

follow the evolution of information-type percepts since they are first perceived until they are forgotten by the simulated user. When they are first perceived (step 4 from the ‘Perception’ section), the simulated user will set up their ‘is old’ component to FALSE, to indicate that they are new percepts. Also, both for the percepts and their attributes, she will set up the ‘remembrance time’ component to 0, to mark that they have just been remembered, and the rest of their persistence components to their default values: ‘in use’ to 0, ‘remembered’ to FALSE and ‘from external’ to FALSE.

Once actions are performed on the device, the simulated user updates the remembrance time of all of the information-type percepts apart from percepts which were set as ‘from external’, and their attributes, with the waiting time of the action, and considers them as old percepts (‘is old’ set to FALSE). She also forgets those structural-type percepts from the perceptual buffer. She then perceives the new device state and, in case percepts which are equivalent to any of the current information-type, and not ‘from external’, ones are found, she reminds herself of the older percepts (step 3a from the ‘Perception’ section), by replacing the old percepts with the new ones, where ‘is old’ is set to FALSE and ‘remembrance time’ is again set to 0 for themselves and their attributes. Once perception is complete, the simulated user forgets all of the information-type percept attributes by removing them from the percept’s set of attributes, if:

- They are not in use (their ‘in use’ component is 0) and have exceeded the simulated user’s maximum remembrance time.
- They are in use (their ‘in use’ component is greater than 0) and have exceeded the simulated user’s maximum remembrance time for information in use.

In case all of the information-type attributes of a percept are thus forgotten, and the percept is not itself in use (its ‘in use’ is equal to 0), the simulated user also forgets the percept by removing it from the perceptual buffer. The next step is to also forget information-type, not ‘from external’ percept attributes, if too many of them have been remembered. The simulated user will give priority to forgetting the attributes which are not in use, and their percepts if they are not in use and are left without any information-type attributes. Then, if this is not enough, the simulated user will also forget attributes which are in use, starting from the oldest remembered ones (i.e. which have the highest ‘remembrance time’). Similarly, if a percept is left without any information-type attributes, and it is not itself in use, the simulated user also forgets the percept.

Whenever a set of information-type percepts are matched with a skill to form a binding and given this binding the skill is accomplishable, the simulated user will remind herself of any remembered (i.e. having 'is old' equal to TRUE) information-type percepts and their information-type attributes which matched to the skill's percepts to form its binding. To this end, she will:

- Reset each such percept's 'remembrance time' component to 0, if the percept was remembered (i.e. has the 'is old' component set to TRUE). I have made the assumption that, when using a percept within a skill, the simulated user always reminds herself of that percept. Even if the percept has defined attributes, her remembrance of them also implies the remembrance of the percept (i.e. of its existence).
- Reset the 'remembrance time' component for any attributes of the matching percept which were required by the skill, if they (and the percept) were remembered (i.e. their 'is old' is set to TRUE)
- Increment the value of their 'in use' component if the attributes have the 'remembered' component set to FALSE (i.e. they have not been previously required to be remembered, as I will explain in the next section), to show that they are being used by a skill. The 'in use' value of these attributes will therefore increase with the number of skills/subskills in which they will be used.
- Increment the percept's 'in use' component if the percept does not have any attributes (i.e. the skill required its existence only) and has the 'remembered' component set to FALSE (i.e. it has not been previously required to be remembered, as I will explain in the next section). As for attributes, 'in use' will thus increase with the number of skills/subskills where the percept's existence only is required.

The percepts from the intention that is instantiated for the skill will have all of their persistence components set to their default values, apart from 'in use' components for attributes, which will be set to 1 to indicate the attributes of the equivalent real percepts which have had their 'in use' component increased as being required by the skill (and to 0 otherwise).

Whenever an intention, be it an instance of a device or external skill, is considered as achieved, the simulated user will decrease the 'in use' components of the real percepts which are equivalent to its percepts, and of their attributes, as guided by the

intention's 'in use' value for them (i.e. if the intention's value is equal to 1, the equivalent real percept's component should be decreased). Also, it will set their 'remembered' component to FALSE, as will be explained in the next section.

E.0.3.4.2 The Mechanism for Percepts to Remember

The users whom I have observed for the real telehealthcare system described in Chapter 3 sometimes needed to look for information as a result to going to a certain page in the website, for using it for a later activity. For example, COs often needed to remember information about alerts from the Alerts page in order to know how to reply to questions from the spreadsheet, from a different tab. I made the assumption that such information would be remembered for the same duration as information which is in use by a skill.

To this end, the simulated user can use 'percepts to remember', which were briefly mentioned in the description of skills. Percepts to remember are an optional component of skills, which could appear in their head or be defined for their subskills. They specify the format of certain percepts which the simulated user would like to remember once the skill, or subskill, is accomplished, with the purpose of using them for another skill. Put into context, we could use them to say that the simulated user's skill of going to the Alerts page should have as a purpose the remembrance of the percepts of type "alert", and with certain attributes, which are needed later. If the real percepts respecting the format of the percepts to remember are perceived once the skill/subskill is accomplished, these percepts will have their 'in use' component as increased to make them more difficult for the simulated user to forget, and their 'remembered' component set to TRUE to indicate that they were remembered for being used by another skill (in this case, the skill of filling in the questions from the spreadsheet). Please be reminded that percepts or attributes which have an 'in use' value greater than 0 will be remembered for longer than other percepts.

When the percepts set up this way become part of the binding for a skill (the first encountered one, e.g. filling in the questions from the spreadsheet), their 'remembered' component, as set to TRUE, will indicate that they were remembered from before, and that 'in use' has already been increased for them. The new skill will therefore, unlike for other percepts, not set them as in use any longer. 'In use' will be decreased, and 'remembered' set to FALSE, when the intention instantiating the skill is achieved.

Percepts to remember look like percepts, without any of the persistence attributes.

They nevertheless differ in terms of the values of their attributes to the percepts from the skill's 'percepts' field. If used within the head of a skill, they should have these values as fully defined as literal or numerical values. In this case, once the skill is instantiated into an intention, they would be used to map against the real percepts having those exact values for their attributes, and conceptually the skill would always require those specific percepts. For example, a skill of going to the Alerts page could require that three of the percepts of type "alert" which have the attribute with type "status" and value "open" be remembered once the skill is accomplished. The skill's head would in this case be represented as follows:

Head: (go_to_alerts ?p ((alert website ?a1 (status Open) TRUE)(alert website ?a2 (status Open) TRUE)(alert website ?a3 (status Open) TRUE))

If used for a subskill (which refers to the head of another skill), they should have the values of their attributes set either as literal or numerical values, variables referring to the values of attributes from the parent skill's 'percepts', or the literal keyword value 'any'. For example, let's assume that the device skill of filling in a question on the spreadsheet requires as the first subskill for the simulated user to go to the Alerts page to retrieve a percept of type "alert", the "text" attribute of which matches to the text attribute of the question, which has the status "Open" and can contain any colour attribute. In this case, the device skill would be represented as follows:

Head: (fill_in_next_question ?p)

Percepts: ((patient website ?p TRUE)(question spreadsheet ?q (text ?t) TRUE))

Conditions: (question_open ?q)

Subskills: ((go_to_alerts ?p (alert website ?a ((text ?t)(status Open)(colour any))))...)

When instantiating the intention for the subskill, any variables for the attribute values of the percepts to remember are replaced by the real attribute values of the bound real percepts from the parent skill. In the example from above, if the simulated user has a percept of type "question" and with the "Systolic value exceeded" value for the attribute "text" which matches to the skill, then the percept of type "alert" that the subskill would be intended to remember would need to have the "text" attribute also set to "Systolic value exceeded". Any literal or numerical attribute values for percepts to remember would need to match with the attribute values of real percepts. In case of the

‘any’ keyword, a percept having an attribute with the required type and with any value can match with the percept to remember, the purpose here being that the attribute of that certain type be present in the matching percept. To sum up, the following would be two percepts that could match the percept to remember from our example, if the value of the “text” attribute of the percept of type “question” is “Systolic value exceeded”:

```
(alert website a1 ((text "Systolic value exceeded")(status Open)(colour red)))
(alert website a1 ((text "Systolic value exceeded")(status Open)(colour orange)))
```

The following percepts would not match:

```
(alert website a1 ((text "Diastolic value exceeded")(status Open)(colour red)))
(alert website a1 ((text "Systolic value exceeded")(status Closed)(colour orange)))
(alert website a1 ((text "Systolic value exceeded")(status Closed)))
```

One limitation of percepts to remember is that they cannot require an unbounded number of matching percepts respecting a certain format. The modeller must provide the definition of each of the percepts.

E.0.3.4.3 The (Infinite-time) Remembrance of Information as an Effect to an External Skill

I observed that a few of the users of the telehealthcare system that I described in Chapter 3 prefer to write information from the telemonitoring system down on paper. In this case, this information is always available to them, they cannot forget it and they do not need to go back through the website to remind themselves of it.

To deal with this situation, I allow the modeller to set up a percept’s attributes or the percept itself as having been observed from the environment, by setting up their ‘from external’ component to TRUE. In the case of writing down on paper, the modeller could set up those specific percept attributes which are being written down on paper, or the percept itself if she writes down only about its existence, this way. For example, she could set up the “type” and “colour” attributes of the percepts of type “alert” as ‘from external’ TRUE if this is what a human would write down on paper. Once this is done, the percepts holding at least one attribute set up as ‘from external’, and all of their

attributes which were set up this way, will never be forgotten ⁴. Their ‘remembrance time’ component cannot increase value as the simulated user takes new actions on the device, and thus they can never be forgotten as having exceeded the maximum remembrance time. When information-type percepts or attributes are being forgotten because of their number, these percepts and attributes, even if information-type, cannot. This does not also apply to other attributes of the percept, which have ‘from external’ equal to FALSE, and which can be forgotten by the simulated user. Also, to cater for the possibility for the values of the attributes which are ‘from external’ to change on the duration of skill execution, I still consider that the simulated user would update her memory of them. This could correspond to her also correcting her observations from the sheet of paper, in our example.

E.0.3.4.4 Other Additions

As I do not have time associated to perception, and users sometimes spend some time on reading information from a system, I have decided to add this time, where it is clear from my observations, by means of external skills, the name of which contains the text “retrieve”. I then realised that, in contrast to other external skills, which lead the simulated user to updating her remembrance time of her information-type percepts and attributes, her reading of the percepts and attributes from the device state as part of a “retrieve” external skill should require the opposite. More exactly, the simulated user should keep their ‘remembrance time’ to 0. I am therefore considering retrieval as a special case for the updating of remembrance time for external skills, where the simulated user only updates the remembrance time for remembered percepts, and not also for newly perceived ones.

The writing of information as part of a device or external skill, or the selecting of information from a dropdown list as part of a device skill, are other special cases, as working with the information would require in their case its continuous perception, and thus remembering. Whenever the simulated user is carrying out skills with the names “write” or “select”, she will update the remembrance time for all of the new and remembered percepts, except from the ones which were used as part of the skill, for which she will keep the remembrance time as 0.

Please note that for both of these cases, those percepts and their attributes which

⁴I assumed here that the fact that the simulated user remembers about an attribute means that she also remembers about the existence of the percept

were set as ‘from external’ true will still have their ‘remembrance time’ kept to 0.

E.0.4 Behaviour of the Device Model

When it receives a requirement for an action from the simulated user, in the form of a user action, the device first looks for its equivalent device action(s). A device action is equivalent to a user action if the device action’s device state components have the same type and attribute names with the percepts having matched the arguments for the user action. If the device does not identify an equivalent device action, it will return an error message. Otherwise, it creates a list with all of the equivalents, and will try to perform them in order to change the device state until one works.

To this end, it will match each device action’s device state components with the real device state components from its state, using the percept names and attribute values required by the user action, to create a binding. This is similar to how percepts from a concept’s ‘percepts’ component are matched with real percepts for conceptual inference by the simulated user. It then does the same for matching the device state components from device conditions, adding new bindings to the binding to fully define the device action. Once defined, the device will first compute the duration of the resulting device action. The duration is a normal random number generated by using either the regular mean and standard deviation, or the one with interruptions, depending on the chance of interruptions, and omitting the results which are less than the lower bound. The device then attempts to change the device state by using the resulting device action.

The step of changing the device state must be defined by the modeller within the implementation, by specifying exactly which device state components must be removed, added or changed as an effect to the device action. The adding of any device state component to the device state will incur a check of whether a device state component with the same name and version have not already been added, in which case the device returns an error message.

Once the device state is changed, the device will return it to the user through the controller.

E.0.5 Behaviour of the Controller

The Controller helps with the communication between the user and device model. It sets up the two models with the inputs provided by the modeller, and sequentially launches cognitive cycles for the user model, and user requests for transitions in the device state

for the device. More exactly, the controller requests the simulated user to do the next step in accomplishing her goals and waits for her reply in the form of a user action that she needs performed on the device. It then sends this user action to the device, and waits for the device to change its state. If the user action is repeatable, it will request the simulated user to repeatedly check the achievement of her intention given the new device state. If not, it requests the simulated user to only do one check. Any of the steps could result in an error message from the user or device model, which would terminate the run. The cycle is repeated until the simulated user has no more steps to perform as having already achieved all of her goals.

As kept conceptually from my previous modelling approach, the controller also checks mode confusion problems. In particular, the controller checks if simulated user expectations from the “effects” of the intentions instantiated from her device skills are met in her beliefs and, if not, if this was due to the simulated user not having perceived well from the system. The first problem can only happen if the user model is changed to not check the effects of intentions obtained from device skills (which may be reasonable for modelling a very experienced user). The second can only happen if a different perception behaviour is implemented for the user model. With the functionality as it is at the time being, they are only useful for detecting modeller mistakes in building the models.

Appendix F

Detailed Requirements for Model Inputs in My Modelling Approach

F.1 Inputs for the User Model

LTM constructs and goals need to be defined in XML files. Their level of detail should be decided on depending on the purposes of the simulation.

The details of repeatable goals need to be provided in the ‘repeatable_goals.txt’ text file. Each line should contain the name of a repeatable goal, the name of the XML file from where to retrieve the goal’s arguments, and an XPath query for accessing them from this XML file. For example: *monitored patients.xml /patients/patient/@id* can be used to specify that ‘monitored’ is a goal which can be called repeatedly, each time with a different patient id, the value of which is taken from ‘patients.xml’, as its argument. Please be reminded that a repeatable goal should only be defined in ‘goals.xml’ with one argument- either ‘any’ or ‘all’.

Both the path of the XML input files for the user and device, and that of the ‘repeatable_goals’ text file need to be provided by the modeller within the Configs class.

To specify the STM capacity, the modeller needs to provide an integer value for the ‘max_information _remembered’ variable from the Configs Java class of the Utils package. For the STM duration, the modeller needs to provide in the same class a higher number for ‘max_in_use _rem_time’ (remembrance time for information that the simulated user wants to remember), and a lower one for ‘max_rem_time’ (remembrance time for information that she does not).

F.2 Inputs for the Device Model

The modeller needs to define the initial device state and the whole set of device actions in XML files, and provide a path for them within the Configs class. The initial device state must only include structural device state components. The modeller should define how the device is to be populated with information-type device state components by implementing the populateInfo method from the DeviceState class of the main package, using the inputs for the run of the models. Also, the computeDeviceEnd method of the TransitionManager class of the main package should be used to define how each action on the device changes the device state. The modeller should therefore have some basic experience with programming in Java, or get help from a Java programmer.

Appendix G

List of Settings

noPatientsCOPD = 17
noPatientsCHF = 3
noCOPDPathA = 4
noCOPDPathD = 3
noCOPDPathF = 1
noCOPDPathG = 3
noCOPDPathH = 4
noCOPDPathI = 2
noCHFPathA = 1
noCHFPathC = 1
noCHFPathG = 1
percentageCallsReplied = 78.57
timeToAnswerMean = 9440
timeToAnswerSD = 4732.155957
timeToAnswerMaxValue = 30000
noCOPDPQuestion1 = 0
noCOPDPQuestion3 = 0
noCOPDPQuestion5 = 0

Appendix H

Glossary

ACT-R: Adaptive Control of Thought-Rational
CHF: chronic heart failure
CMN-GOMS: Card, Moran and Newell GOMS
CO: call operator
COPD: chronic obstructive pulmonary disease
CPA: Critical Path Analysis
CPM: Critical Path Method
CPM-GOMS: Cognitive Perceptual Motor, or Critical Path Method GOMS
CRT: Community Respiratory Team
DALLAS: Delivering Assisted Living Lifestyles at Scale
DLPFC: dorsolateral prefrontal cortex
EMACS: East and Midlothian COPD Team
fMRI: functional magnetic resonance imaging
GOMS: Goals, Operators, Methods and Selection Rules
GP: general practitioner
GUI: graphical user interface
HCI: Human Computer Interaction
ICT: Information and Communication Technologies
IDE: integrated development environment
IEC: International Electrotechnical Commission
ISO: International Organization for Standardization
ITTS: Implementing Transnational Telemedicine Solutions
JDK: Java Development Kit
KLM: Keystroke-Level Model

LTM: long-term memory
LTS: labelled transition system
LUCS: Lothian Unscheduled Care Service
MHP: Model Human Processor
NGOMSL: Natural GOMS Language
NHS: National Health System
OS: operating system
PERT: Program Evaluation and Review Technique
PUM: Programmable User Models
STM: short term Memory
TSB: Technology Strategy Board
VLPFC: ventroLateral preFrontal cortex
WM: working memory
XML: Extensible Markup Language

Bibliography

- [1] Implementing Transnational Telemedicine Solutions (ITTs). URL <http://www.transnational-telemedicine.eu/>.
- [2] Living It Up. URL <https://portal.livingitup.org.uk/page/about-us>.
- [3] Math Is Fun: Relative Frequency, . URL <https://www.mathsisfun.com/data/relative-frequency.html>.
- [4] Math Is Fun: Standard Deviation Formulas, . URL <http://www.mathsisfun.com/data/standard-deviation-formulas.html>.
- [5] The Free Dictionary: telemetry. URL <http://medical-dictionary.thefreedictionary.com/telemetry>.
- [6] Modelling and Simulation Overview. URL <http://acqnotes.com/acqnote/tasks/modeling-simulation-overview>.
- [7] REgioNs of Europe WorkINg toGether for HEALTH (Renewing Heath). URL <http://www.renewinghealth.eu/en/home>.
- [8] BS 3138:1992 Glossary of terms used in management services, April 1992. URL <http://shop.bsigroup.com/ProductDetail/?pid=000000000000265244>.
- [9] The future of health care and care for the elderly: guaranteeing accessibility, quality and financial viability. Communication from the European Commission, European Commission, December 2001. URL <http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52001DC0723&from=EN>.
- [10] ACT-R Perceptual-Motor, July 2006. URL <http://chil.rice.edu/projects/RPM/index.html>.
- [11] Telemedicine for the benefit of patients, healthcare systems and society. European Commission Staff Working Paper, European Commission, June 2009. URL ec.europa.eu/information_society/newsroom/cf/document.cfm?action=display&doc_id=622.
- [12] Introduction to Modeling and Simulation, 2009. URL <http://www.acqnotes.com/Attachments/Introduction%20to%20Modeling%20and%20Simulation.pdf>.

- [13] Projected Population of Scotland (2010-based). Technical report, National Records of Scotland, 2011. URL <http://www.nrscotland.gov.uk/statistics-and-data/statistics/statistics-by-theme/population/population-projections/population-projections-scotland/2010-based>.
- [14] eHealth strategy 2011-2017. Technical report, The Scottish Government, September 2011. URL <http://www.scotland.gov.uk/Publications/2011/09/09103110/9>.
- [15] DALLAS - Delivering Assisted Living Lifestyles at Scale, June 2011. URL http://gallery.mailchimp.com/7872a44a98b8760bc45fdf56c/files/sbri_comp_dallas_sbri_final.pdf.
- [16] Commission Staff Working Document on the applicability of the existing EU legal framework to telemedicine services. European Commission Staff Working Paper, European Commission, December 2012. URL http://ec.europa.eu/information_society/newsroom/cf/document.cfm?doc_id=1251.
- [17] Digital health care, November 2013. URL <http://news.scotland.gov.uk/News/Digital-health-care-628.aspx>.
- [18] A Universal Truth: No Health Without a Workforce. Technical report, World Health Organisation, 2013. URL http://www.who.int/workforcealliance/knowledge/resources/GHWA_AUniversalTruthReport.pdf.
- [19] Blossom: Home Blood Pressure Monitoring In Pregnancy Using Telemetry, 2014. URL <http://www.telescot.org/blossom.html>.
- [20] The impact of a telemetric monitoring service in diabetes type 2, 2014. URL <http://www.telescot.org/diabetes.html>.
- [21] Telescot: investigating telemetric supported self monitoring of long term health conditions, 2014. URL <http://www.telescot.org/>.
- [22] What is Template Analysis?, 2014. URL <http://www.hud.ac.uk/hhs/research/template-analysis/what-is-template-analysis/>.
- [23] DALLAS - Delivering Assisted Living Lifestyles at Scale, 2015. URL <https://connect.innovateuk.org/web/dallas/about-us1>.
- [24] A. Abran, A. Khelifi, W. Suryn, and A. Seffah. Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal*, pages 325–338, 2003.
- [25] C.A. Alexandru. Usability Testing and Improvement of Telemedicine Websites. Master's thesis, University of Edinburgh, School of Informatics, 2010. URL <http://www.inf.ed.ac.uk/publications/thesis/online/IM100870.pdf>.

- [26] C.A. Alexandru. Using Models and Simulation for Predicting Efficiency as a Measure of Success of Different Telemedicine Deployments. In *Proceedings of eTELEMED 2014 The Sixth International Conference on eHealth, Telemedicine, and Social Medicine*, pages 230–235, Barcelona, Spain, March 23–27 2014. URL http://www.thinkmind.org/index.php?view=article&articleid=etelemed_2014_11_30_40216.
- [27] C.A. Alexandru and B. McKinstry. Usability Evaluation of Clinician Web Back-Ends to Telemonitoring Systems: Two Case-Studies in Scotland. *Studies in Informatics and Control*, 21(2):181–190, 2012. URL http://sic.ici.ro/?page_id=389.
- [28] C.A. Alexandru and P. Stevens. Models to Extend the Scope of Usability Testing for Telemedicine Systems. In *Digital Futures 2012: The Third Annual Digital Economy All Hands Conference*, Aberdeen, 2012. URL http://www.dotrural.ac.uk/digitalfutures/sites/default/files/digitalfutures2012papers/Papers/Session4CUsability/Alexandru%26Stevens_TelemedicineSystems.pdf.
- [29] C.A. Alexandru and P. Stevens. Predicting the Usability of Telemedicine Systems in Different Deployments through Modelling and Simulation. In *Proceedings of the European Workshop on Practical Aspects of Health Informatics (PAHI 2013)*, Edinburgh, 2013. URL <http://ceur-ws.org/Vol-984/paper6.pdf>.
- [30] D. Alonso-Ros, A. Vázquez-García, E. Mosqueira-Rey, and V. Moret-Bonillo. Usability: A Critical Analysis and a Taxonomy. *International Journal Human-Computer Interaction*, 26(1):53–74, 2010.
- [31] E.M. Altmann. Near-term memory in programming: A simulation-based analysis. *International Journal of Human-Computer Studies*, 54(2):189–210, 2001.
- [32] E.M. Altmann and B.E. John. Episodic indexing: A model of memory for attention events. *Cognitive Science*, 23(2):117–156, 1999.
- [33] J.R. Anderson. Using brain imaging to guide the development of a cognitive architecture. In W.D. Gray, editor, *Integrated Models of Cognitive Systems*, pages 49–62. Oxford University Press, 2007.
- [34] J.R. Anderson and C. Lebiere. *The Atomic Components of Thought*. Erlbaum, June 1998. ISBN 0805828176.
- [35] J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglas, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111:1036–1060, 2004.
- [36] M. Anu. Introduction to Modeling and Simulation. In *Proceedings of the 1999 Winter Simulation Conference*, pages 7–13. IEEE Computer Society, 1997.
- [37] R.C. Atkinson and R.M. Shiffrin. Human memory: A proposed system and its control processes. In K.W. Spence and J. T. Spence, editors, *The psychology of learning and motivation*, volume 2, pages 89–195. Academic Press, New York, 1968.

- [38] A.D. Baddeley, N. Thomson, and M. Buchanan. Word length and the structure of short-term memory. *Journal of Verbal Learning and Verbal Behavior*, 14(6): 575-589, December 1975. doi: 10.1016/S0022-5371(75)80045-4.
- [39] J. Barlow, D. Singh, S. Bayer, and R. Curry. A systematic review of the benefits of home telecare for frail elderly people and those with long-term conditions. *Journal of Telemedicine and Telecare*, 13(4):172-179, 2007.
- [40] R.M. Barnes. *Motion and Time Study*. John Wiley, fourth edition edition, 1958.
- [41] John B.E. Baumeister, L.K. and M.D. Byrne. A comparison of tools for building GOMS models. In NY: ACM Press New York, editor, *Proceedings of the Conference on Human Factors in Computing Systems*, volume 1, page 502-509, The Hague, The Netherlands, April 2000.
- [42] B.J. Best and C. Lebiere. Cognitive agents interacting in real and virtual worlds. In R. Sun, editor, *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, pages 186–218. Cambridge University Press, 2006.
- [43] N. Bevan, J. Kirakowski, and J. Maissel. What is Usability. In *Human Aspects in Computing: Design and Use of Interactive Systems with Terminals*, pages 651–655. Elsevier, 1991.
- [44] R. Bias. Interface-Walkthroughs: efficient collaborative testing. *Software, IEEE*, 8(5):94–95, September 1991. ISSN 0740-7459. doi: 10.1109/52.84220.
- [45] R. Bias. Pluralistic usability walkthrough: coordinated empathies. In *Usability Inspection Methods*, pages 63–76. Wiley and Sons, 1994.
- [46] A. Blandford, R. Butterworth, and J. Good. Users as rational interacting agents: formalising assumptions about cognition and interaction. In *DSV-IS*, pages 45–60, 1997.
- [47] A. Blandford, R. Butterworth, and P. Curzon. Models of interactive systems: a case study on programmable user modelling. *International Journal of Human-Computer Studies*, 60(2):149–200, 2004.
- [48] D. Bloom, A. Boersch-Supan, P. McGee, and A. Seike. Population Aging: Facts, Challenges, and Responses. Pgda working paper no. 71, May 2011. URL <http://core.ac.uk/download/pdf/6494803.pdf>.
- [49] J.A. Borges, I. Morales, and N.J. Rodriguez. Guidelines for designing usable World Wide Web pages. In ACM: New York, editor, *CHI-96 Conference on Human Factors in Computing Systems: Common Ground*, page 277-278, 1996.
- [50] G.H. Bower. Mental imagery and associative learning. In L. Gregg, editor, *Cognition in Learning and Memory*, pages 51–88. 1972.
- [51] T.H. Broens, R.M. Huis in't Veld, M.M. Vollenbroek-Hutten, H.J. Hermens, A.T. van Halteren, and L.J. Nieuwenhuis. Determinants of successful telemedicine implementations: a literature study. *Journal of Telemedicine and Telecare*, 13(6): 303–309, September 2007. doi: 10.1258/135763307781644951.

- [52] C. Bromley, S. Dowling, L. Gray, S. Hinchliffe, T. Hughes, A. Leyland, J. Mabelis, and H. Wardle. The Scottish Health Survey. A national statistics publication for scotland, The Scottish Government, 2012.
- [53] T.A. Burke, J.R. McKee, and H.C. Wilson. A comparison of time-and-motion and self-reporting methods of work measurement. *Journal of Nursing Administration*, 30:118–125, 2000.
- [54] R. Butterworth and A. Blandford. Programmable user models: the story so far. Technical report, 1997. URL <http://www.eis.mdx.ac.uk/puma/wp8.pdf>.
- [55] R. Butterworth, A. Blandford, and D. Duke. The role of formal proof in modelling interactive behaviour. In P. Markopulos and P. Johnson, editors, *Proceedings on Design, Specification and Verifications of Interactive Systems 98*, pages 87–101. Springer, 1998.
- [56] M. Byrne. ACT-R/PM and menu selection: applying a cognitive architecture to HCI. *International Journal Human-Computer Studies*, 2001. doi: 10.1006/ijhc.2001.0469.
- [57] M.D. Byrne. Cognitive architecture. In J. Jacko and A. Sears, editors, *Human-Computer Interaction Handbook*, chapter Cognitive architecture, pages 97–117. Lawrence Erlbaum Associates, 2003.
- [58] Wood S.D. Sukaviriya P.N. Foley J D. Byrne, M.D. and D Kieras. Automating interface evaluation. In NY: ACM Press New York, editor, *Proceedings of the Conference on Human Factors in Computing Systems*, volume 1, page 232237, April 1994.
- [59] D.G. Cacuci. *Sensitivity and Uncertainty Analysis*, volume 1. Chapman & Hall/CRC, 2003.
- [60] J. Car, K. Huckvale, and H. Hermens. Telehealth for long term conditions. *BMJ: British Medical Journal*, June 2012. doi: <http://dx.doi.org/10.1136/bmj.e4201>.
- [61] S.K. Card, T.P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Communications. ACM*, 23:396–410, July 1980. ISSN 0001-0782.
- [62] S.K. Card, A. Newell, and T.P. Moran. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1983. ISBN 0898592437.
- [63] A. Careaga. Eye-tracking studies: first impressions form quickly on the web, February 2012. URL http://news.mst.edu/2012/02/eye-tracking_studies_show_firs/.
- [64] M. Cartwright, S.P. Hirani, L. Rixon, M. Beynon, H. Doll, P. Bower, M. Bardsley, A. Steventon, M. Knapp, C. Henderson, A. Rogers, R. Fitzpatrick, J. Barlow, and S.P. Newman. Effect of telehealth on quality of life and psychological outcomes

- over 12 months (Whole Systems Demonstrator telehealth questionnaire study): nested study of patient reported outcomes in a pragmatic, cluster randomised controlled trial. *BMJ: British Medical Journal*, 346(653):1–20, February 2013. doi: doi:http://dx.doi.org/10.1136/bmj.f653.
- [65] A. Chandor, J. Graham, and R. Williamson. *Dictionary of Computers (Reference Books)*. Penguin Books Ltd, 1985.
 - [66] S.I. Chaudhry, C.O. Phillips, S.S. Stewart, B. Riegel, J.A. Mattera, and A.F. Jerant. Telemonitoring for patients with chronic heart failure: a systematic review. *Journal of Cardiac Failure*, 13:56–62, February 2007.
 - [67] D. Choi, M. Kaufman, P. Langley, N. Nejati, and D. Shapiro. An architecture for persistent reactive behavior. In New York: ACM Press, editor, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 988–995, 2004.
 - [68] D. Choi, T. Könik, N. Nejati, C. Park, and P. Langley. A Believable Agent for First-Person Shooter Games. In *AIIDE*, pages 71–73, 2007.
 - [69] C. Connelly. NHS has 'more pilots than British Airways'. *The Guardian*, May 2011. URL <http://www.theguardian.com/healthcare-network/2011/may/12/christine-connelly-nhs-more-pilots-british-airways>.
 - [70] R. Conrad and A.J. Hull. Information, Acoustic Confusion and Memory Span. *British Journal of Psychology*, 55(4):429432, November 1964.
 - [71] W.D. Cottrell. Simplified program evaluation and review technique (PERT). *Journal of construction Engineering and Management*, 125(1):16–22, 1999.
 - [72] J. Cruickshank. Healthcare without Walls: Delivering telehealth at scale, November 2010. URL <http://www.2020health.org/2020health/Publications/Publications-2010/telehealth.html>.
 - [73] A. Crystal and B. Ellington. Task analysis and human-computer interaction: approaches, techniques, and levels of analysis. In *Proceedings of the Tenth Americas Conference on Information Systems*, New York, August 2004.
 - [74] R. Currell, C. Urquhart, P. Wainwright, and R. Lewis. *Telemedicine Versus Face to Face Patient Care: Effects on Professional Practice and Health Outcomes*. Cochrane Library, issue 3 edition, 2002.
 - [75] G. Demiris and D. Tao. An analysis of the specialized literature in the field of telemedicine. *Journal of Telemedicine and Telecare*, 11(6):316, 2005.
 - [76] A. Deshpande, S. Khoja, J. Lorca, A. McKibbin, C. Rizo, and A.R. Jadad. Asynchronous Telehealth Systematic Review of Analytic Studies and Environmental Scan of Relevant Initiatives. Technical report, Canadian Agency for Drugs and Technologies in Health, Ottawa, 2008.

- [77] A. Deshpande, S. Khoja, A. McKibbin, and A.R. Jadad. Real-Time (synchronous) Telehealth in Primary Care: Systematic Review of Systematic Reviews, 2008.
- [78] A. Dix, J. Finlay, G.D. Abowd, and R.L. Beale. *Human- Computer Interaction, Third Edition*. Pearson Education Limited, 2004.
- [79] J.S. Dumas and J.E. Fox. Usability testing: current practice and future directions. In Andrew Sears and Julie E. Jacko, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Lawrence Erlbaum Associates, Inc., 2008.
- [80] A.G. Ekeland, A. Bowes, and S. Flottorp. Effectiveness of telemedicine: A systematic review of reviews. *International Journal of Medical Informatics*, 79: 736771, October 2010.
- [81] A.G. Ekeland, A. Bowes, and S. Flottorp. Methodologies for assessing telemedicine: A systematic review of reviews. *International Journal of Medical Informatics*, 81:1–11, 2012.
- [82] P Fairbrother, J. Ure, J. Hanley, L. McCloughan, M. Denvir, A. Sheikh, and B. McKinsty. Telemonitoring for chronic heart failure: the views of patients and healthcare professionals a qualitative study. *Journal of Clinical Nursing*, 2012. doi: doi:10.1111/jocn.12137.
- [83] A. Farmer, O.J. Gibson, L. Tarassenko, and A. Neil. A systematic review of telemedicine interventions to support blood glucose self-monitoring in diabetes. *Diabetic Medicine*, 22 .:13721378, October 2005.
- [84] J.R. Finkler, S.A .and Knickman and G. Hendrickson. A comparison of work-sampling and time-and-motion techniques for studies in health services research. *Health Services Research Journal*, 1993.
- [85] M.D. Fleetwood and M. Byrne. Modeling the visual search of displays: A revised ACT-R/PM model of icon search based on eye-tracking and experimental data. In *Proceedings of the Fifth International Conference on Cognitive Modeling*, pages 87–92. Universitas-Verlag Bamberg Bamberg, Germany, 2003.
- [86] M.D. Fleetwood and Bryne M.D. Modeling icon search in act-r/pm. In E.M. Altmann, Cleermans A., Schunn C. D., and Gray W.D., editors, *Fourth International Conference on Cognitive Modeling (ICCM 2001)*, pages 17–22, Mahwah, NJ, 2001. Earlbaum.
- [87] E. Folmer and J. Bosch. Cost Effective Development of Usable Systems: Gaps between HCI and Software Architecture Design. In *Advances in Information Systems Development*, pages 337–348. Springer US, 2006. doi: 10.1007/978-0-387-36402-5_29.
- [88] R. Gaikwad and J. Warren. The role of home-based information and communications technology interventions in chronic disease management: a systematic literature review. *Health Informatics Journal*, 15(2):122146, June 2009.

- [89] F. Garcia-Lizana and A. Sarria-Santamera. New technologies for chronic disease management and control: a systematic review. *Journal of Telemedicine and Telecare*, 13(2):6268, 2007.
- [90] F.B. Gilbreth. Scientific management in the hospital. In *Modern Hospital*, volume 3, page 321324. 1914.
- [91] W.D. Gray, B.E. John, and M.E. Atwood. Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance. *Humuman-Computer Interaction*, 8:237–309, September 1993. ISSN 0737-0024.
- [92] ACT-R Research Group. About ACT-R, 2013. URL <http://act-r.psy.cmu.edu/about/>.
- [93] D. Hailey, R. Roine, and A. Ohinmaa. Systematic review of evidence for the benefits of telemedicine. *Journal of Telemedicine and Telecare*, 8(Suppl. 1):130, 2002.
- [94] H.R. Hartson, H.C. Castillo, and J. Kelso. Remote Evaluation: The Network as an Extension of the Usability Laboratory. In *CHI96 Conference Proceedings*, 1996.
- [95] C. Henderson, M. Knapp, H.-L. Fernndez, J. Beecham, S.P. Hirani, M. Cartwright, L. Rixont, M. Beynon, A. Rogers, P. Bower, H. Doll, R. Fitzpatrick, A. Steventon, M. Bardsley, J. Hendy, and S.P. Newman. Cost effectiveness of telehealth for patients with long term conditions (Whole Systems Demonstrator telehealth questionnaire study): nested economic evaluation in a pragmatic, cluster randomised controlled trial. *BMJ: British Medical Journal*, 346(1035):1–19, March 2013. doi: 10.1136/bmj.f1035.
- [96] W.R. Hersh, M. Helfand, J. Wallace, D. Kraemer, P. Patterson, S. Shapiro, and M. Greenlick. Clinical outcomes resulting from telemedicine interventions: a systematic review. *BMC Medical Informatics and Decision Making*, 1(1):5, 2001.
- [97] W.R. Hersh, D. Hickam, S. Severance, T. Dana, K. Krages, and M. Helfand. Diagnosis, access and outcomes: update. *Journal of Telemedicine and Telecare*, 12(2):331, 2006.
- [98] W.R. Hersh, D.H. Hickam, S.M. Severance, T.L. Dana, K.P. Krages, and M. Helfand. Telemedicine for the medicare population: update, Evid. Rep. Technol. Assess. (Full Rep.). *AHRQ Publication*, 131(06-E007):141, February 2006.
- [99] M. Hertzum and N.E. Jacobsen. The Evaluator Effect: A Chilling Fact About Usability Evaluation Methods. *International Journal Human-Computer Interaction*, 13(4):421–443, 2001.
- [100] F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*. McGraw-Hill International Editions. McGraw-Hill, 2001. ISBN 9780072321692.

- [101] G. Hitchcock. Do we need a telehealth tsar? *The Guardian*, December 2012. URL <http://www.theguardian.com/healthcare-network/2012/dec/12/telehealth-care-pathways-results-nhs-direct>.
- [102] N.M. Hjelm. Benefits and drawbacks of telemedicine. *Journal of Telemedicine and Telecare*, 11(2):60–70, March 2005.
- [103] T. Hollingsed and D.G. Novick. Usability inspection methods after 15 years of research and practice. In *Proceedings of the 25th annual ACM international conference on Design of communication*, SIGDOC '07, pages 249–255, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-588-5. doi: <http://doi.acm.org/10.1145/1297144.1297200>.
- [104] J. Hom. The Usability Methods Toolbox, 1998. URL <http://usability.jameshom.com/index.htm>.
- [105] A. Howes and R.M. Young. Learning consistent, interactive, and meaningful taskaction mappings: A computational model. *Cognitive Science*, 20(3):301–356, 1996.
- [106] John B.E. Knudsen K. Hudson, S.E. and M. Byrne. A tool for creating predictive performance models from user interface demonstrations. In NY: ACM Press New York, editor, *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, page 92103, Asheville, NC, November 1999.
- [107] ISO. ISO 9241-11, 1998. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883.
- [108] ISO/IEC. ISO/IEC 9126 Software engineering - Product quality – Part 1: Quality model, 2001. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749.
- [109] ISO/IEC. ISO/IEC 25010:2011 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, 2011. URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733.
- [110] M.Y. Ivory and M.A. Hearst. The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys*, 33:470–516, 2001.
- [111] M. Jaana, G. Pare, and C. Sicotte. Home telemonitoring for respiratory conditions: a systematic review. *American Journal of Managed Care*, 15(5):313320, May 2009.
- [112] C.L. Jackson, S. Bolen, F.L. Brancati, M.L. Batts-Turner, and T.L. Gary. A systematic review of interactive computer-assisted technology in diabetes care. interactive information technology in diabetes care. *Journal of General Internal Medicine*, 21(2):105110, February 2006.

- [113] N.E. Jacobsen, M. Hertzum, and B.E. John. The evaluator effect in usability tests. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 255–256, New York, NY, USA, 1998. ACM. doi: 10.1145/286498.286737.
- [114] M.W. Jaspers. A comparison of usability methods for testing interactive health technologies: Methodological aspects and empirical evidence. *International Journal of Medical Informatics*, 78(5):340–353, 2009.
- [115] R. Jeffries and H. Desurvire. Usability testing vs. heuristic evaluation: was there a contest? *ACM SIGCHI Bulletin*, 24(4):39–41, 1992.
- [116] R. Jeffries, J. Miller, C. Wharton, and K. Uyeda. User interface evaluation in the real world: a comparison of four techniques. In S.P. Robertson, G.M. Olson, and J.S. Olson, editors, *Proc. CHI91*, page 119124. ACM Press, New York, 1991.
- [117] P. Jennett, L. Affleck Hall, D. Hailey, A. Ohinmaa, C. Anderson, R. Thomas, B. Young, D. Lorenzetti, and R.E. Scott. The socio-economic impact of telehealth: a systematic review. *Journal of Telemedicine and Telecare*, 9(6):311, 2003.
- [118] JIT. A National Telehealth and Telecare Delivery Plan for Scotland to 2015. Technical report, NHS Scotland, 2012. URL www.scotland.gov.uk/Resource/0041/00411586.pdf.
- [119] B.E. John. Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90, pages 107–116, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6. doi: <http://doi.acm.org/10.1145/97243.97262>.
- [120] B.E. John. Information Processing and Skilled Behaviour. In John M. Carroll, editor, *HCI models, theories, and frameworks- Toward a multidisciplinary science*, chapter 4, pages 55–101. Morgan Kaufmann, 2003.
- [121] B.E. John and D. Kieras. Using GOMS for user interface design and evaluation: which technique? *ACM Transactions of Computer-Human Interaction*, 3:287–319, December 1996. ISSN 1073-0516.
- [122] R.M. Jones, J.E. Laird, P.E. Nielsen, K.J. Coulter, P. Kenny, and F.V. Koss. Automated intelligent pilots for combat flight simulation. *Artificial Intelligence Magazine*, 20 (1):27–41, 1999.
- [123] M. Kahn and A. Prail. Formal usability inspections. In *Usability inspection methods*, pages 141–171. John Wiley & Sons, New York, 1994.
- [124] B.T. Karsh. Beyond usability: designing effective technology implementation systems to promote patient safety. In *Quality and Safety in Health Care*, volume 13, pages 388–394. 2004.
- [125] Jr. Kelley, E. James, and M.R. Walker. Critical-path planning and scheduling. In *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, pages 160–173. ACM, 1959.

- [126] D. Kieras. Towards a practical GOMS model methodology for user interface design. In M. Helander, editor, *Handbook of human-computer interaction*, pages 135–157. Elsevier Science, North-Holland, 1988.
- [127] D. Kieras. A Guide to GOMS Task Analysis, 1994.
- [128] D. Kieras. Model-based evaluation. In *Human-Computer Interaction: Development Process*, Human Factors and Ergonomics, chapter 15, pages 293–311. Sears, A. and Jacko, J. A., 2009.
- [129] Wood S.D. Abotel K. Kieras, D.E. and A. Hornof. GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface design. In NY: ACM Press New York, editor, *Proceedings of the Eighth ACM Symposium on User Interface Software and Technology*, page 91100, Pittsburgh, PA, November 1995.
- [130] M. Kipp. ANVIL - The video annotation research tool, 2012. URL <http://www.anvil-software.org>.
- [131] K.R. Koedinger, J.R. Anderson, W.H. Hadley, and M. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8(1):30–43, 1997.
- [132] M. Kykk, R. Ollikainen, M. Ranta-Aho, W. Milszus, S. Wasseroth, and M. Friedrich. Usability heuristic guidelines for 3D multi-user worlds. In *Proc. CHI99 Conference, New South Wales, Australia*, 1999.
- [133] J.E. Laird. Extending the Soar cognitive architecture. In *Artificial General Intelligence Conference*, 2008.
- [134] J.E. Laird, A. Newell, and P.S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.
- [135] S. Laja. Telehealth hindered by lack of clinical evidence, says academic. *The Guardian*, July 2011. URL <http://www.theguardian.com/healthcare-network/2011/jul/04/telehealth-hindered-lack-clinical-evidence-academic>.
- [136] T.K. Landauer. *The Trouble With Computers: Usefulness, Usability, and Productivity*. MIT Press (A Bradford Book), Cambridge, MA, 1995.
- [137] P. Langley and D. Choi. A unified cognitive architecture for physical agents. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2, AAAI'06*, pages 1469–1474. AAAI Press, 2006. ISBN 978-1-57735-281-5.
- [138] P. Langley and S. Rogers. An extended theory of human problem solving. In *Proceedings of the Twenty-Seventh Annual Meeting of the Cognitive Science Society*, Stresa, Italy, 2005.

- [139] P. Langley, K. Cummings, and D. Shapiro. Hierarchical skills and cognitive architectures. In *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society*, pages 779–784, 2004.
- [140] P. Langley, D. Choi, and S. Rogers. Acquisition of hierarchical reactive skills in a unified cognitive architecture. *Cognitive Systems Research*, 10(4):316–332, 2009.
- [141] P. Langley, J.E. Laird, and S. Rogers. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10(2):141–160, 2009.
- [142] P. Langley, D. Choi, and N. Trivedi. ICARUS User’s Manual, July 2011. URL http://www.dongkyu.com/icarus-tutorial/ICARUS_manual_3.pdf.
- [143] J.F. Lehman, J.E. Laird, and P. Rosenbloom. A Gentle Introduction to Soar, an Architecture for Human Cognition: 2006 Update, 2006. URL <http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf>.
- [144] C. Lewis, P.G. Polson, C. Wharton, and J. Rieman. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI ’90, pages 235–242, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6. doi: <http://doi.acm.org/10.1145/97243.97279>.
- [145] R.L. Lewis. An architecturally-based theory of sentence comprehension. In *Proceedings of the fifteenth annual conference of the cognitive science society*, pages 108–113. Lawrence Erlbaum, 1993.
- [146] X. Lia and M. Gunalb. Exploring cognitive modelling in engineering usability design. *Journal of Engineering Design*, 23(2):77–97, 19 Nov 2012. doi: 10.1080/09544828.2010.528379.
- [147] M. Lopetegui, P.Y. Yen, A. Lai, J. Jeffries, P. Embi, and P. Payne. Time motion studies in healthcare: What are we talking about? *Journal of Biomedical Informatics*, 49:292299, 2014.
- [148] B. Magerko, J.E. Laird, M. Assanie, A. Kerfoot, and D. Stokes. Ai characters and directors for interactive computer games. In *Proceedings of the sixteenth innovative applications of artificial intelligence conference*, pages 877–884. AAAI Press, 2004.
- [149] M. Maguire. Methods to support human-centred design. *International Journal Human-Computer Studies*, 55:587–634, 2001.
- [150] D.G. Malcolm, J.H. Roseboom, C.E. Clark, and W. Fazar. Application of a technique for research and development program evaluation. *Operations research*, 7(5):646–669, 1959.
- [151] T. Martand. *Industrial Engineering and Production Management*. Chand, S., New Dehly, 2000.

- [152] H.B. Maynard, J.L. Schwab, and G.J. Stegemerten. *Methods Time Measurement*. McGraw-Hill, 1948.
- [153] E. McCann. Study challenges telehealth benefits. *Healthcare IT News*, May 2013. URL <http://www.healthcareitnews.com/news/study-challenges-telehealth-benefits>.
- [154] B. Mckinstry, H. Pinnock, and A. Sheikh. Telemedicine for management of patients with COPD? *The Lancet*, 374(9691):672–673, September 2009. doi: 10.1016/S0140-6736(09)61542-7.
- [155] B. McKinstry, J. Hanley, S. Wild, C. Pagliari, M. Paterson, S. Lewis, A. Sheikh, A. Krishan, A. Stoddart, and P. Padfield. Telemonitoring based service re-design for the management of uncontrolled hypertension: multicentre randomised controlled trial. *BMJ: British Medical Journal*, 346, May 2013. doi: <http://dx.doi.org/10.1136/bmj.f3030>.
- [156] C.S. Miller and J.E. Laird. Accounting for graded performance within a discrete search framework. *Cognitive Science*, 20:499–537, 1996.
- [157] G.A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):8197, 1956. doi: 10.1037/h0043158.
- [158] D.Z. Milosevic. *Project management toolbox: tools and techniques for the practicing project manager*. John Wiley & Sons, 2003.
- [159] R. Molich, M.R. Ede, and K. Kaasgaard. Comparative Usability Evaluation. *Behaviour and Information Technology*, 23:65–74, 2004.
- [160] R. Molich, R. Jeffries, and J. Dumas. Making usability recommendations useful and usable. *Journal of Usability Studies*, 2(4):162–179, August 2007.
- [161] E. Murphy, R. Dingwall, D. Greatbatch, S. Parker, and P. Watson. Qualitative research methods in health technology assessment: a review of the literature. In *Health Technol. Assess.*, volume 2, chapter III-IX, pages 1–274. 1998.
- [162] G. Nelson, J.F. Lehman, and B.E. John. Integrating cognitive capabilities in a real-time task. In A. Ram and K. Eiselt, editors, *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pages 353–358. Erlbaum, 1994.
- [163] A. Newell. The knowledge level. In *Artificial Intelligence*, volume 18, pages 87–127. 1982.
- [164] A. Newell. *Unified theories of cognition*. Harvard University Press, Cambridge, MA, USA, 1990. ISBN 0-674-92099-6.
- [165] A. Newell and H.A. Simon. GPS: A program that simulates human thought. In Edward A. Feigenbaum and Jerome A. Feldman, editors, *Computers and Thought*. McGraw-Hill, New York, 1963.

- [166] Allen Newell, Herbert Alexander Simon, et al. *Human problem solving*, volume 104. Prentice-Hall Englewood Cliffs, NJ, 1972.
- [167] D. Nicolini. The work to make telemedicine work: A social and articulative view. *Social science & medicine*, 62(11):2754–2767, 2006.
- [168] J. Nielsen. Usability engineering at a discount. In G. Salvendy and M. J. Smith, editors, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, pages 394–401. Elsevier Science Publishers, Amsterdam, 1989.
- [169] J. Nielsen. Finding usability problems through heuristic evaluation. In *Proc. CHI'92 Conf.*, pages 373–380, Monterey, CA, May 3-7 1992. ACM, New York.
- [170] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Pub, 1993.
- [171] J. Nielsen. Ten Usability Heuristics, January 1995. URL www.useit.com/papers/heuristic/heuristic_list.html.
- [172] J. Nielsen. Discount Usability: 20 Years, September 2009. URL <http://www.useit.com/alertbox/discount-usability.html>.
- [173] J. Nielsen. Usability 101: Introduction to Usability, January 2012. URL <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [174] J. Nielsen and R. Molich. Heuristic Evaluation of User Interfaces. In *Proceedings of ACM CHI90 Conference on Human Factors in Computing Systems*, pages 249–256. Elsevier Science Publishersb, 1990.
- [175] N.J. Nilsson. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill Pub. Co., 1971. ISBN 0070465738.
- [176] D.A. Norman. *The Design of Everyday Things*. Currency and Doubleday, New York, 1988.
- [177] D.A. Norman. *The Design of Everyday Things*. Basic Books, New York, reprint paperback edition, 2002. ISBN 0-465-06710-7.
- [178] J.R. Olson and G.M. Olson. The growth of cognitive modeling in human-computer interaction since GOMS. *Human-Computer Interaction*, 5:221–265, June 1990. ISSN 0737-0024. doi: http://dx.doi.org/10.1207/s15327051hci0502\&3_4.
- [179] C. Pagliari. Design and Evaluation in eHealth: Challenges and Implications for an Interdisciplinary Field. *Journal of Medical Internet Research*, 9(2), 2007.
- [180] E.W. Patton, W.D. Gray, and B.E. John. Automated CPM-GOMS Modeling from Human Data. In *Proceedings of the 56th Annual Meeting of the Human Factors and Ergonomics Society*, Boston, MA, October 22-26 2012.

- [181] H. Pinnock, J. Hanley, L. McCloughan, A. Todd, A. Krishan, S. Lewis, A. Stoddart, M. van der Pol, W. MacNee, A. Sheikh, C. Pagliari, and B. McKinstry. Effectiveness of telemonitoring integrated into existing clinical services on hospital admission for exacerbation of chronic obstructive pulmonary disease: researcher blind, multicentre, randomised controlled trial. *BMJ: British Medical Journal*, 347, October 2013. doi: <http://dx.doi.org/10.1136/bmj.f6070>.
- [182] P. Pirolli and W. Fu. SNIF-ACT: A Model of Information Foraging on the World Wide Web. In *Proceedings of the Ninth International Conference on User Modeling*, pages 45–54. Springer-Verlag, 2003.
- [183] M. Poirier and J. Saint-Aubin. Memory for related and unrelated words: Further evidence concerning the influence of semantic factors on immediate serial recall. *Quarterly Journal of Experimental Psychology*, 48A(2):384–404, 1995.
- [184] M. Poirier and J. Saint-Aubin. Immediate serial recall, word frequency, item identity and item position. *Canadian Journal of Experimental Psychology*, 50: 408–412, 1996.
- [185] P. Polson, C. Lewis, J. Rieman, and C. Wharton. Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, 36:741–773, 1992.
- [186] J. Preece, H. Rogers, Y. and Sharp, D. Benyon, S. Holland, and T. Carey. *Human-Computer Interaction*. Addison Wesley, 1994.
- [187] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, New York, 2002.
- [188] B. Price. Frank and Lillian Gilbreth and the Motion Study Controversy, 1970–1973. In Daniel Nelson, editor, *A Mental Revolution: Scientific Management since Taylor*. The Ohio State University Press, 1990.
- [189] J. Rieman, R.M. Young, and A. Howes. A dual-space model of iteratively deepening exploratory learning. *International Journal of Human-Computer Studies*, 44(6):743–775, 1996.
- [190] F.E. Ritter, D. Van Rooy, and R.St. Amant. A User Modeling Design Tool Based on a Cognitive Architecture for Comparing Interfaces. In C. Kolski and J. Vanderdonckt, editors, *Computer-Aided Design of User Interfaces III, Proceedings of the 4th International Conference on Computer-Aided Design of User Interfaces CADUI'2002*, pages 111–118. Kluwer, Academic Publisher, Dordrecht., 2002.
- [191] F.E. Ritter, D. Van Rooy, R.St. Amant, and K. Simpson. Providing user models direct access to interfaces: An exploratory study of a simple interface with implications for HRI and HCI. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 36(3):592–601, 2006.

- [192] R. Roine, A. Ohinmaa, and D. Hailey. Assessing telemedicine: a systematic review of the literature. *Canadian Medical Association Journal*, 165.(6):765, 2001.
- [193] S.V. Rojas and M.P. Gagnon. A systematic review of the key indicators for assessing telehomecare cost-effectiveness. *Telemedicine and E Health*, 14(9):896904, November 2008.
- [194] Johnny Saldaña. *The coding manual for qualitative researchers*. Number 14. Sage, 2012.
- [195] D.L. Sanders, D. Aronsky, and J. Am. Biomedical informatics applications for asthma care: a systematic review. *Medical Informatics Association*, 13(4): 418427, JulyAugust 2006.
- [196] Health Informatics Scotland. Telehealth, 2010. URL <http://www.knowledge.scot.nhs.uk/his/new-to-health-informatics/telehealth.aspx>.
- [197] NHS Scotland. Improving Health and Wellbeing of People with Long Term Conditions in Scotland: A National Action Plan, June 2009. URL <http://www.gov.scot/Resource/Doc/294270/0090939.pdf>.
- [198] A. Sears. Heuristic walkthroughs: finding the problems without the noise. *International Journal Human-Computer Interaction*, 9.(3):213234, 1997.
- [199] A. Seffah, M. Donyaee, R.B. Kline, and H.K. Padda. Usability measurement and metrics: A consolidated model. In *Software Quality Control*, volume 14, page 159178. 2006.
- [200] E. Seto. Cost comparison between telemonitoring and usual care of heart failure: a systematic review. *Telemedicine Journal and E-Health*, 14(7):679686, September 2008.
- [201] B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Publ. Co., Reading, MA, fifth edition, 2010.
- [202] J. Singh, C. Lutteroth, and B.C. Wunsche. A Taxonomy of Usability Requirements for Home Telehealth Systems. In *Proceedings of the 11th International Conference of the NZ Chapter of the ACM Special Interest Group on Human-Computer Interaction*, CHINZ '10, pages 29–32, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0104-6. doi: <http://doi.acm.org/10.1145/1832838.1832844>.
- [203] W. Stevenson. *Operations Management (Operations and Decision Sciences)*. McGraw-hill Publishing House, 11 edition, 2011.
- [204] A. Steventon, M. Bardsley, J. Billings, J. Dixon, H. Doll, S. Hirani, M. Cartwright, M. Knapp, C. Henderson, A. Rogers, R. Fitzpatrick, J. Hendy, and S. Newman. Effect of telehealth on use of secondary care and mortality: findings from the

- Whole System Demonstrator cluster randomised trial. In *BMJ: British Medical Journal*, pages 1–15. June 2012. doi: <http://dx.doi.org/10.1136/bmj.e3874>.
- [205] A. Steventon, M. Bardsley, and J. Billings. Effect of telecare on use of health and social care services: findings from the Whole Systems Demonstrator cluster randomised trial. *BMJ: British Medical Journal*, June 2013.
- [206] D. Svanæs, A. Das, and O.A. Alsos. The Contextual Nature of Usability and its Relevance to Medical Informatics. In *MIE*, pages 541–546, 2008.
- [207] F.W. Taylor. *The Principles of Scientific Management*. Harper & Brothers, New York, 1911.
- [208] J.G. Trafton, N.L. Cassimatis, M.D. Bugajska, D.P. Brock, F.E. Mintz, and A.C. Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 35:460–470, 2005.
- [209] K. Tran, J. Polisen, and D. Coyle. Home Telehealth for Chronic Disease Management. Technology Report No 113 113, Canadian Agency for Drugs and Technologies in Health (CADTH), 2008.
- [210] M. Tsiknakis and A. Kouroubali. Organizational factors affecting successful adoption of innovative eHealth services: A case study employing the FITT framework. *International Journal of Medical Informatics*, In Press, Corrected Proof, 2009. doi: 10.1016/j.ijmedinf.2008.07.001.
- [211] M. van Welie, G.C. van der Veer, and A. Elins. Breaking down Usability. In *Proceedings of Interact 99*, pages 613–620, Edinburgh, Scotland, 30th August - 3rd September 1999.
- [212] F. Verhoeven, L. van Gemert-Pijnen, K. Dijkstra, N.E. Nijland, Seydel E., and M. Stehouder. The contribution of teleconsultation and videoconferencing to diabetes care: a systematic literature review. *Journal of Medical Education Internet Research*, 9(5), 2007.
- [213] D. Vinegar. Let's stop purely academic delays to telehealth. *The Guardian*, July 2011. URL <http://www.theguardian.com/healthcare-network/2011/jul/18/purely-academic-delays-telehealth-patient-hell>.
- [214] P. Vora and M. Helander. A teaching method as an alternative to the concurrent think-aloud method for usability testing. In *Symbiosis of Human and Artifact*, pages 375–380. Y. Anzai, K. Ogawa & H. Mori, 1995.
- [215] C. Wharton, J. Rieman, C. Lewis, and P. Poison. The cognitive walkthrough method: A practitioners guide. In J. Nielsen and R. L. Mack, editors, *Usability Inspection Methods*, pages 105–140. John Wiley & Sons, 1994.
- [216] P. Whitten, F. Mair, A. Haycox, C. May, T. Williams, and S. Hellmich. Systematic review of cost effectiveness studies of telemedicine interventions. *BMJ: British Medical Journal*, 324(7351):1434, 2002.

- [217] Morgen Witzel. *Fifty key figures in management*. Routledge, 2003.
- [218] D. Wixon, S. Jones, L. Tse, and G. Casaday. Inspections and design reviews: Framework, history, and reflection. In *Usability Inspection Methods*, pages 79–104. John Wiley & Sons, 1994.
- [219] WSD. Whole System Demonstrator Programme - Headline Findings. Technical report, Gov UK, December 2011. URL <https://www.gov.uk/government/publications/whole-system-demonstrator-programme-headline-findings-december-2011>.
- [220] R.M. Young, T.R.G. Green, and T. Simon. Programmable user models for predictive evaluation of interface designs. *SIGCHI Bulletin*, 20:15–19, March 1989. ISSN 0736-6906. doi: <http://doi.acm.org/10.1145/67450.67453>.
- [221] L. Yuanhua, A.L. Osvalder, and S. Dahlman. Exploring user background settings in cognitive walkthrough evaluation of medical prototype interfaces: a case study. *International Journal of Industrial Ergonomics*, 35:379390, 2005.
- [222] Z. Zhang. Usability Evaluation. URL <http://www.usabilityhome.com/>.
- [223] Z. Zhang. Usability Evaluation. In Sri Kurniawan Panayiotis Zaphiris, editor, *Human Computer Interaction Research in Web Design and Evaluation*, chapter X, pages 209–228. IGI, 2007.
- [224] K. Zheng, M. Guo, and D. Hanauer. Using the time and motion method to study clinical work processes and workflow: methodological inconsistencies and a call for standardized research. *Journal of the American Medical Informatics Association*, 18:704710, 2011.